



# TMx00 APPLICATION NOTE

## CORRECTION ALGORITHM DEVELOPMENT FOR CHARACTERIZATION AND MANUFACTURING

RFA-04-042 Rev 1.2

06/2023

### Hexius Semiconductor Proprietary

**Restricted Distribution.** Not to be distributed to anyone who is not an employee of Hexius Semiconductor without the express approval of Hexius Semiconductor's management.

Not to be used, copied, reproduced in whole or in part, nor its contents revealed in any manner to others without the express permission of Hexius Semiconductor.

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited.

Hexius Semiconductor  
4640 E. Elwood Street, Suite 19  
Phoenix, AZ 85040  
U.S.A.

Copyright © 2023 Hexius Semiconductor  
All rights reserved.

# CONTENTS

Contents .....	2
Figures .....	3
Tables.....	3
Revision History .....	4
1 Correction Algorithm Overview.....	5
Adjustable Timing Parameters .....	6
Lookup Table Curve Fit .....	6
Temperature Correction Polynomial Curve Fit .....	7
Supply Voltage Curve Fit.....	8
2 Correction Algorithm Development Flow .....	10
Disable TMx00 Correction .....	11
Set Environmental Chamber to Initial Temperature .....	12
<i>Collecting Single Point Temperature Data</i> .....	12
Step Environmental Chamber Through Temperature Range.....	13
<i>Process Data Determine Correction Parameters</i> .....	13
Program the TMx00 & Validation.....	13
3 Temperature Data Collection Flow.....	15
Set Environmental Chamber to Starting Temperature Point .....	15
Measure Temperature & Read <i>Temp Code</i> .....	16
Determine the <i>CorrDAC Code</i> for Center Frequency.....	17
4 Voltage Data Collection Flow .....	18
Set Supply Voltage.....	18
Measure Supply Voltage and Read <i>VDDA Code</i> .....	18
Measure Temperature & Read <i>Temp Code</i> .....	20
5 Lookup Table Implementation Flow.....	21
Select Lookup Table.....	22
Fill TMx100 Lookup Table.....	23
Perform Supply Voltage Correction (Optional) .....	24
6 Temperature Polynomial Implementation Flow .....	25
Compute the <i>CorrDAC Voltages</i> for all Temperature Points.....	26
Compute the Mean, Std Dev, and Normalization of <i>Temp Codes</i> .....	27
Perform Temperature Correction Polynomial Curve Fit for Coefficient Generation .....	28

	Enter Coefficients, Mean, & Standard Deviation Values.....	29
7	Supply Voltage Implementation Flow .....	30
	<i>Temp Code</i> Adjustment Example - Linear Correction .....	32
	<i>CorrDAC</i> Code Adjustment Example - Linear Correction.....	36
8	References .....	38

## FIGURES

Figure 1	<i>Temp Code</i> and <i>CorrDAC Code</i> Relationship .....	5
Figure 2	Linear Lookup Table Concept.....	6
Figure 3	Main Flow for Correction Algorithm Development .....	10
Figure 4	Temperature Data Collection Flow .....	15
Figure 5	Voltage Data Collection Flow .....	18
Figure 6	Lookup Table Implementation Flow .....	21
Figure 7	Linear Interpolation <i>Temp Code</i> Spacing Considerations.....	24
Figure 8	Temperature Polynomial Implementation Flow.....	25
Figure 9	<i>Temp Code</i> & <i>CorrDAC</i> Voltage Over Temperature Sweep .....	26
Figure 10	Supply Voltage Polynomial Implementation Flow.....	30
Figure 11	<i>VDDA</i> Code & <i>Temp</i> Voltage Over Supply Voltage Sweep.....	32

## TABLES

Table 1	Example <i>Temp Code</i> & <i>CorrDAC Code</i> Raw Data Set.....	17
Table 2	Example <i>VDDA Code</i> & <i>Temp Code</i> Raw Data Set.....	20
Table 3	<i>CorrDAC</i> Voltage Calculation Example .....	26
Table 4	Example with Normalized Temperature Code .....	27
Table 5	Data Values Entered Into Polynomial Curve Fit Calculation .....	28
Table 6	Temperature Correction Polynomial Coefficients.....	28
Table 7	Voltage Correction <i>Temp Code</i> Data Set Example .....	32
Table 8	Voltage Correction <i>CorrDAC Code</i> Data Set Example .....	36

## REVISION HISTORY

Revision	Date	Description
1.0	01/2023	Initial Release
1.1	04/2023	Updated for Firmware 1.8
1.2	06/2023	Updated for Firmware 1.9

# 1 CORRECTION ALGORITHM OVERVIEW

The TMx00 supports three correction algorithms to compensate for temperature and voltage non-idealities:

1. **Lookup Table** – Temperature correction algorithm
2. **Temperature Polynomial Curve Fit** – Temperature correction algorithm
3. **Supply Voltage Curve Fit** – Voltage correction algorithm

The TMx00 correction algorithms use integer numbers (digital codes) that represent temperatures and voltages to make compensation adjustments. The ADC converts analog signals into the digital domain for the MCU to calculate the appropriate Correction DAC input code and produce an analog correction voltage.

The first two algorithms use a *Temp Code* input to generate a *CorrDAC Code* from the Correction DAC output to correct frequency variations over temperature. Only one of the temperature correction algorithms may be used during operation.

The *Temp Code* value is an integer with the range of 0 to 4095 and corresponds to the temperature being measured and digitized through the ADC. A -40C to 90C temperature range will utilize a *Temp Code* range of approximately 2000 to 3200. The *Temp Code* is produced from either the IC internal temperature sensor or an external thermistor (via the THRM pin).

The *CorrDAC Code* value is an integer with the range of 0 to 4095 and corresponds to the DAC input code needed to vary the capacitance across an external varactor or the internal TMx00 Varicap to correct the frequency variation for a given temperature.

The *CorrDAC Voltage* is resulting Correction DAC output voltage for a given *CorrDAC Code*. It is calculated by multiplying the Correction DAC's reference voltage by the ratio of the *CorrDAC Code* to the Correction DAC's full-scale code (4095).

$$\text{CorrDAC Voltage} = \frac{\text{CorrDAC Code}}{4095} * \text{BYPASS Pin Voltage}$$

As a brief relationship example, a measured temperature of 25C may produce a *Temp Code* of 2541 and results in a *CorrDAC Code* of 1983 and a *CorrDAC Voltage* of 1.4043V for the correct center frequency of a unique TCXO assembly.

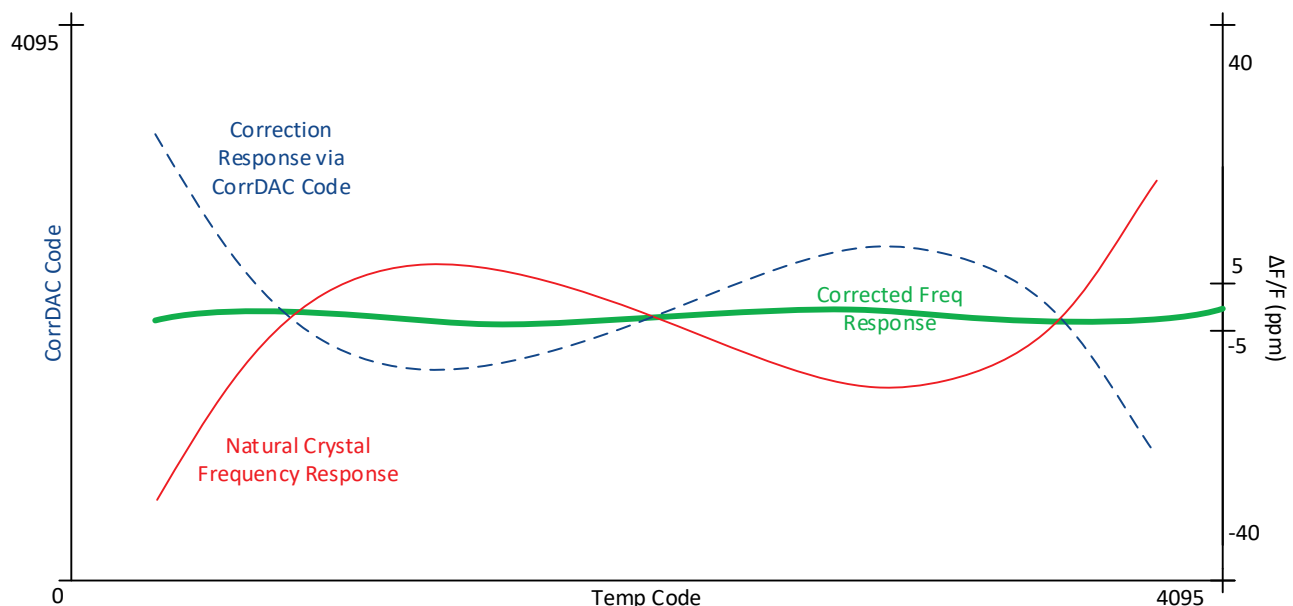


Figure 1 *Temp Code* and *CorrDAC Code* Relationship

## Adjustable Timing Parameters

The TMx00 correction mechanism is a hybrid analog/digital implementation and has several adjustable timing parameters that allow customers to tune timing modules to the time constants of the intended applications. The adjustments also implement low pass filters to reduce crystal adjustment transients.

Supported adjustable timing parameters include:

**Temperature Sample Interval** – This sets the time interval (in ms) for measuring the temperature sensor and VDDA value. The interval is between 20ms and 2550ms. Both the temperature and VDDA readings are averaged over 8 sample intervals, updated at each sample interval (rolling average).

**Frequency Correction Interval** - This sets the time interval (in ms) for updating the Correction DAC value applied to the Varicap or external varactor. The interval is between 20ms and 2550ms.

**Averaging Intervals** - This sets the number of Frequency Correction Intervals used to calculate an average DAC correction value, updated each correction interval (rolling average). The averaging is between 1 and 16.

## Lookup Table Curve Fit

The Lookup Table correction technique uses the measured reading from the IC temperature sensor or external thermistor (*Temp Code*) and generates a *CorrDAC Code* via a lookup table.

The lookup table is arranged from the lowest *Temp Code* to the highest *Temp Code* with up to 80 defined points. The spacing between the points is user defined to account for changing crystal temperature coefficient slopes. The appropriate DAC correction code (*CorrDAC Code*) between the defined *Temp Code* points is calculated by linear interpolation.

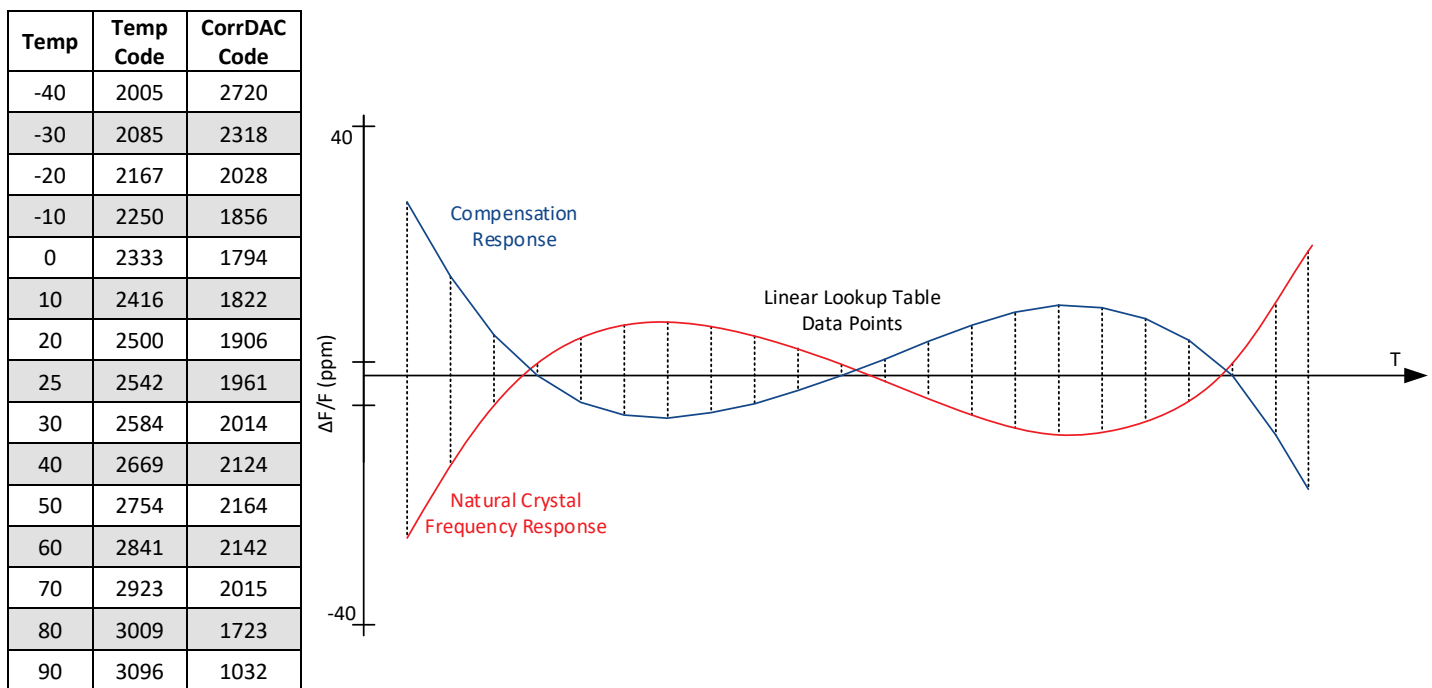


Figure 2 Linear Lookup Table Concept

## Temperature Correction Polynomial Curve Fit

The temperature polynomial curve fit correction technique uses the measured temperature reading from the IC sensor or external thermistor (*Temp Code*) and calculates a *CorrDAC Voltage* via polynomial curve fit using the correction order coefficients,  $a_0 - a_9$ .

The correction process computes the Correction DAC voltage needed for best correction. The *CorrDAC Voltage* range is from 0V up the voltage of the BYPASS pin (typically, 2.9V or 3.3V depending on configuration). The correction value is converted into a 12-bit DAC input digital code (*CorrDAC Code*), 0 for 0V, and 4095 for the max BYPASS Voltage. The internal tuning Varicap operates over a range of 0V to 2.9V, so the *CorrDAC Voltage* needs to be limited when BYPASS is operated from a supply greater than 2.9V.

$$Corr_{temp}(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7 + a_8x^8 + a_9x^9$$

$x = TempCodeNormalized$ : *TempCode normalized for Mean and Standard Deviation*

$$Corr_{temp}(x) = CorrDAC Voltage = Correction DAC voltage$$

To use the polynomial curve fit algorithm, the following data needs to be selected or entered:

1. Desired correction order: For some applications 3<sup>rd</sup> or 5<sup>th</sup> order corrections may produce better results over temperature than a 7<sup>th</sup> or 9<sup>th</sup> order correction. This item can be set as needed for each oscillator application.
2. *Temp Code* Mean and *Temp Code* Std Dev: Measure the module over temperature and capture the *Temp Code* for each measured temperature point. Compute the mean and standard deviation of the *Temp Code* values. Then scale the *Temp Code* values by the mean and standard deviation generating  $x$  (*Temp Code Normalized*) for the above equation.
3. *CorrDAC Codes* over Temperature: As part of the module measurements over temperature, determine the *CorrDAC Code* that produces the least error. Convert the *CorrDAC Code* to *CorrDAC Voltage* by scaling it, 0 to 4095 for the voltage range 0 to the BYPASS pin Voltage. *CorrDAC Voltage* is  $Corr_{temp}(x)$  in the equation above.
4. Use polynomial curve fit software to find the values of  $a_0$  to  $a_n$  based on  $x$  and  $Corr_{temp}(x)$ .
5. Enter the coefficients,  $a_0 - a_9$ , the *Temp Code* mean, and the *Temp Code* Std Dev for the respective correction order. The coefficients are single precision floating point numbers stored in the microcontroller. Single precision floating point numbers have 7-8 decimal digits of precision and ensure that all significant digits are entered into the microcontroller.

## Supply Voltage Curve Fit

The *CorrDAC Voltage* values are not overly sensitive to supply voltage (VDDA) variation when the internal BYPASS regulator is enabled. The *Temp Code Normalized* values have some dependency on supply voltage. This means that supply voltage variations will shift the *Temp Code* values. The supply voltage polynomial curve fit correction techniques are optional enhanced correction methods that generate adjustment terms to account for this shift.

A *Corr<sub>voltage</sub>* term corrects for frequency shifts over supply voltage variation and uses the coefficients,  $b_1 - b_3$ .

A *Corr<sub>voltage\_temp</sub>* term corrects for frequency shifts from cross correlated supply voltage and temperature variation and uses the coefficients,  $c_1 - c_3$ .

The user selects whether these correction terms are applied to either the *Temp Code* or *CorrDAC Code* depending on the architecture and sensitivity of supply voltage movement of the TCXO module. Each choice has further options regarding the order of correction desired.

## Temp Code Adjustment Option

Because the supply voltage range of the TMx00 is  $\pm 5\%$ , the standard deviation of the *VDDA Code* range is set internally with respect to the nominal *VDDA Code*.

$$VDDA_{SD} = \text{Standard deviation of VDDA}$$

The Standard Deviation is defined so a 5% high supply voltage gives a  $VDDA_{SD}$  (or  $\sigma$ ) of +3.0, and a 5% low supply gives a  $VDDA_{SD}$  of -3.0.

$$VDDA_{SD} = VDDACode_{NOM} \left( \frac{0.05}{3} \right)$$

$VDDACode_{NOM}$  is the ADC code value measured by the IC when the VDDA supply voltage is 3.3V. A typical ADC code value is 2330. Note that the VDDA ADC input is divided by 2, so the 2330 code represents an input value of 1.65V.

$$\Delta VDDA = \frac{VDDACode_{Meas} - VDDACode_{Nom}}{VDDA_{SD}}$$

Applying the correction by adjusting the *TempCode*:

$$Temp_{ADJ} = TempCodeNormalized + \Delta TempCode(\Delta VDDA)$$

Where:

$Temp_{ADJ}$  is the new *TempCodeNormalized* value input into the main *Corr<sub>temp</sub>*( $x$ ) function for frequency correction

$\Delta TempCode$  is the *Temp Code* adjustment based on voltage and temperature measurements

*Temp Code Corr<sub>voltage</sub> 1<sup>st</sup> Order Correction Option*

$$\Delta TempCode(\Delta VDDA) = Corr_{voltage}(\Delta VDDA) = b_1(\Delta VDDA)$$

*Temp Code Corr<sub>voltage</sub> 3<sup>rd</sup> Order Correction Option*

$$\Delta TempCode(\Delta VDDA) = Corr_{voltage}(\Delta VDDA) = b_1(\Delta VDDA) + b_2(\Delta VDDA)^2 + b_3(\Delta VDDA)^3$$

*Temp Code Corr<sub>voltage</sub> & Corr<sub>voltage\_temp</sub> 3<sup>rd</sup> Order Correction Option*

$$\begin{aligned} \Delta TempCode(\Delta VDDA) &= Corr_{voltage}(\Delta VDDA) + Corr_{voltage\_temp}(\Delta VDDA) \\ &= b_1(\Delta VDDA) + b_2(\Delta VDDA)^2 + b_3(\Delta VDDA)^3 + c_1(\Delta VDDA)(TempCode) \\ &\quad + c_2(\Delta VDDA)^2(TempCode) + c_3(\Delta VDDA)(TempCode)^2 \end{aligned}$$



**CorrDAC Code Adjustment Option**

Applying the correction by adjusting the *CorrDAC Code*:

$$CorrDAC_{ADJ} = CorrDAC\ Code + \Delta CorrDAC\ Code(\Delta VDDA) - \text{when using the Lookup Table Correction}$$

Or

$$CorrDAC_{ADJ} = CorrDAC\ Voltage + \Delta CorrDAC\ Voltage(\Delta VDDA) - \text{when using the Polynomial Correction}$$

Where:

$CorrDAC_{ADJ}$  is the new *CorrDAC Code* value output for frequency correction (Lookup Table Correction)

$\Delta CorrDAC\ Voltage$  is the new *CorrDAC Voltage* output for frequency correction (Polynomial Correction)

$\Delta CorrDAC\ Code$  is the *CorrDAC Code/Voltage* adjustment based on voltage and temperature measurements.

CorrDAC Code  $Corr_{voltage}$  1<sup>st</sup> Order Correction Option

$$\Delta CorrDAC\ Code(\Delta VDDA) = Corr_{voltage}(\Delta VDDA) = b_1(\Delta VDDA)$$

CorrDAC Code  $Corr_{voltage}$  3<sup>rd</sup> Order Correction Option

$$\Delta CorrDAC\ Code(\Delta VDDA) = Corr_{voltage}(\Delta VDDA) = b_1(\Delta VDDA) + b_2(\Delta VDDA)^2 + b_3(\Delta VDDA)^3$$

CorrDAC Code  $Corr_{voltage}$  &  $Corr_{voltage\_temp}$  3<sup>rd</sup> Order Correction Option

$$\Delta CorrDAC\ Code(\Delta VDDA) = Corr_{voltage}(\Delta VDDA) + Corr_{voltage\_temp}(\Delta VDDA) = b_1(\Delta VDDA) + b_2(\Delta VDDA)^2 + b_3(\Delta VDDA)^3 + c_1(\Delta VDDA)(TempCode) + c_2(\Delta VDDA)^2(TempCode) + c_3(\Delta VDDA)(TempCode)^2$$

## 2 CORRECTION ALGORITHM DEVELOPMENT FLOW

Developing the correction algorithms on the TMx00 is described in four sections:

1. **Main Flow** – High level overview flow of entire development process
2. **Temperature Data Collection Flow** – Details into the data collection performed at each temperature step
3. **Voltage Data Collection Flow** – Details into the data collection performed at each supply voltage step
4. **Correction Implementation Flows** – Details into processing the data required for each correction technique and calculating the needed parameters.

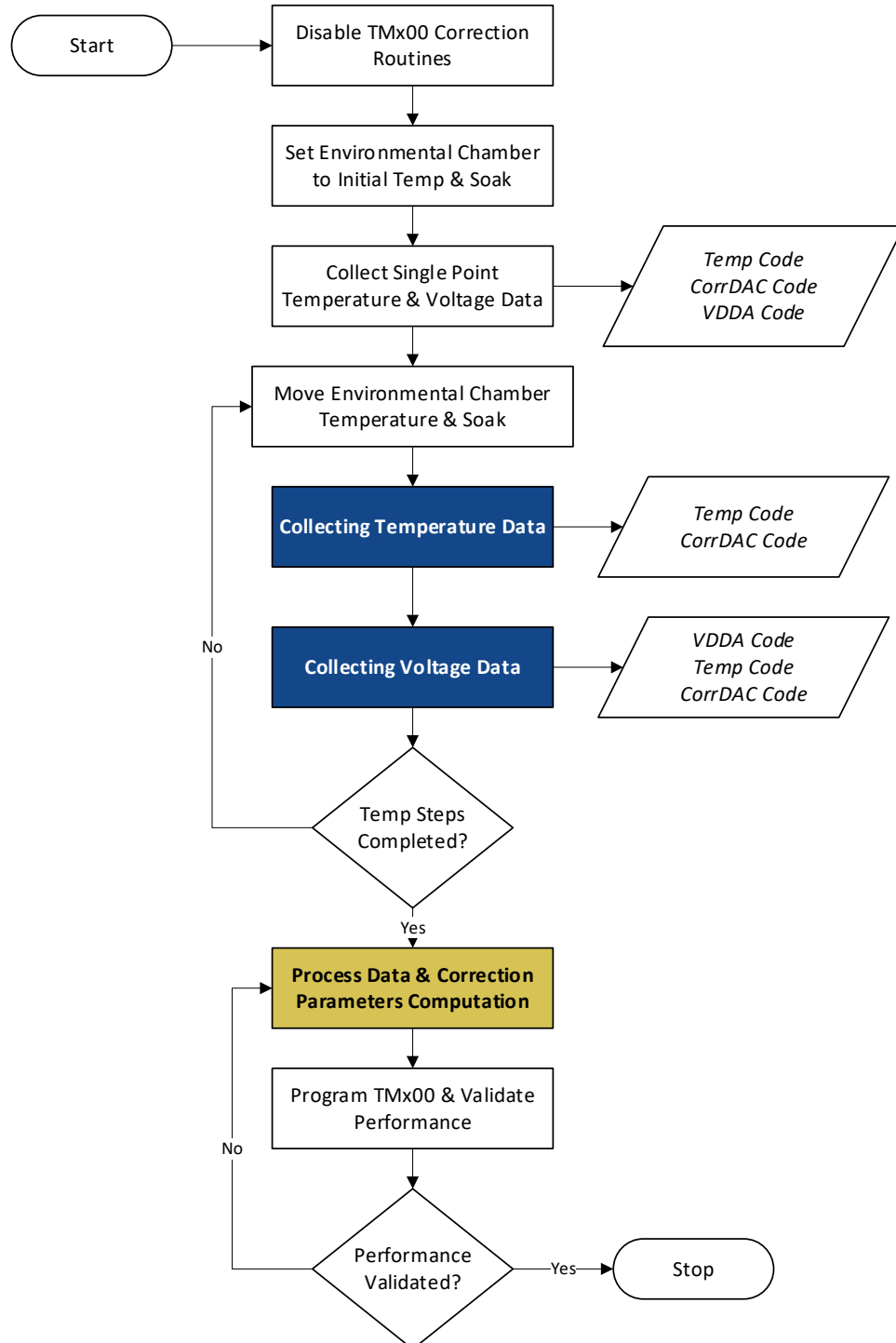
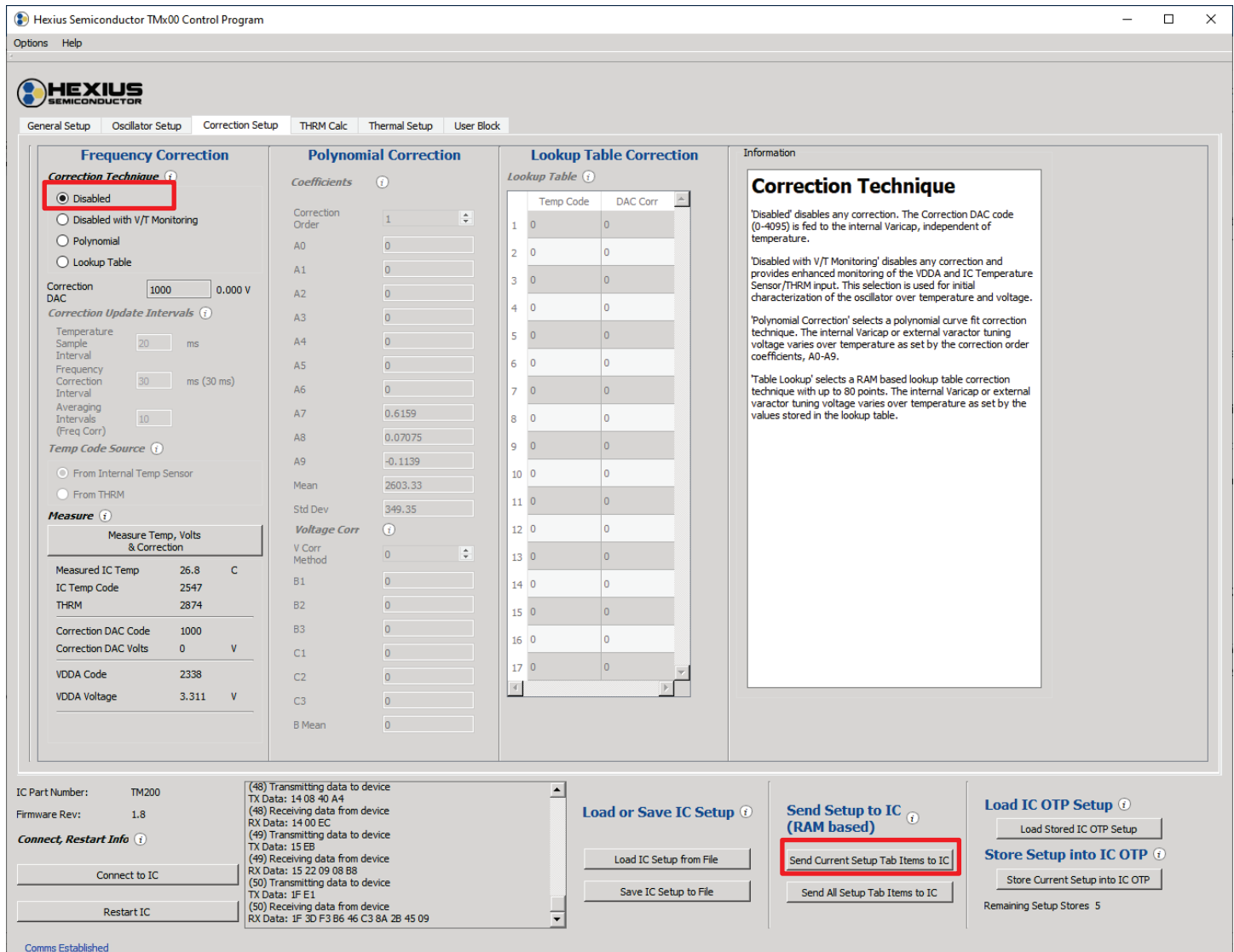


Figure 3 Main Flow for Correction Algorithm Development

## Disable TMx00 Correction

Select the 'Disabled' Correction Technique and send the configuration to the IC by pressing the 'Send Current Setup Tab Items to IC.' If the Correction Technique is not disabled, the IC correction algorithm will be active and interfere with data collection.



## I<sup>2</sup>C Command Option

Instead of using the TMx00 Control Software, the TMx00 correction routines are disabled by writing into the corr\_order RAM setup location with following I<sup>2</sup>C commands (See the *TMx00 Programming Reference Manual* for more information):

Send: 0x19, 0x20, 0x00, 0x01, 0x00, 0xC6

Recv: 0x19, 0x00, 0xE7

Set the corr\_b0 term to 0.0 ensuring no voltage trim adjustments. Since this is a floating-point value, write four bytes.

Send: 0x19, 0xE0, 0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x03

Recv: 0x19, 0x00, 0xE7

## Set Environmental Chamber to Initial Temperature

Set the environmental chamber to the desired initial temperature and soak as appropriate.

## Collecting Single Point Temperature Data

The temperature measurement can be sourced from the internal temperature sensor of the IC (which is a measurement of the IC temperature) or from an external thermistor through the THRM pin of the IC. Select which source to collect the *Temp Code* data from. The internal temperature sensor is the default configuration since it requires no additional external components or setup. Send the configuration to the IC by pressing the 'Send Current Setup Tab Items to IC.'

The Correction DAC input window allows the user to manually enter a *CorrDAC Code* (0 to 4095) and send the value to the IC by pressing the 'Send Current Setup Tab Items to IC' to set the desired frequency. The user can also retrieve the *Temp Code* and *VDDA Code* along with other measured/converted data by pressing the 'Measure Temp, Volts & Correction' button.

Hexius Semiconductor TMx00 Control Program

Options Help

General Setup Oscillator Setup **Correction Setup** THRM Calc Thermal Setup User Block

**Frequency Correction**

**Correction Technique**

- ☒ Disabled
- ☐ Disabled with V/T Monitoring
- ☐ Polynomial
- ☐ Lookup Table

Correction DAC: 1000 0.000 V

**Correction Update Intervals**

Temperature Sample Interval: 20 ms

Frequency Correction Interval: 30 ms (30 ms)

Averaging Intervals (Freq Corr): 10

**Temp Code Source**

- ☐ From Internal Temp Sensor
- ☐ From THRM

**Measure**

Measure Temp, Volts & Correction

Measured IC Temp	26.8	C
IC Temp Code	2547	
THRM	2874	
Correction DAC Code	1000	
Correction DAC Volts	0	V
VDDA Code	2338	
VDDA Voltage	3.311	V

**Polynomial Correction**

**Coefficients**

Correction Order: 1

A0: 0

A1: 0

A2: 0

A3: 0

A4: 0

A5: 0

A6: 0

A7: 0.6159

A8: 0.07075

A9: -0.1139

Mean: 2603.33

Std Dev: 349.35

**Voltage Corr**

V Corr Method: 0

B1: 0

B2: 0

B3: 0

C1: 0

C2: 0

C3: 0

B Mean: 0

**Lookup Table Correction**

**Lookup Table**

Temp Code	DAC Corr
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0

**Information**

**Correction Technique**

'Disabled' disables any correction. The Correction DAC code (0-4095) is fed to the internal Varicap, independent of temperature.

'Disabled with V/T Monitoring' disables any correction and provides enhanced monitoring of the VDDA and IC Temperature Sensor/THRM input. This selection is used for initial characterization of the oscillator over temperature and voltage.

'Polynomial Correction' selects a polynomial curve fit correction technique. The internal Varicap or external varactor tuning voltage varies over temperature as set by the correction order coefficients, A0-A9.

'Table Lookup' selects a RAM based lookup table correction technique with up to 80 points. The internal Varicap or external varactor tuning voltage varies over temperature as set by the values stored in the lookup table.

IC Part Number: TM200

Firmware Rev: 1.8

**Connect, Restart Info**

Connect to IC

Restart IC

(48) Transmitting data to device  
TX Data: 14 08 40 A4  
(48) Receiving data from device  
RX Data: 14 00 EC  
(49) Transmitting data to device  
TX Data: 15 EB  
(49) Receiving data from device  
RX Data: 15 22 09 08 B8  
(50) Transmitting data to device  
TX Data: 1F E1  
(50) Receiving data from device  
RX Data: 1F 3D F3 B6 46 C3 8A 2B 45 09

**Load or Save IC Setup**

Load IC Setup from File

Save IC Setup to File

**Send Setup to IC (RAM based)**

Send Current Setup Tab Items to IC

Send All Setup Tab Items to IC

**Load IC OTP Setup**

Load Stored IC OTP Setup

**Store Setup into IC OTP**

Store Current Setup into IC OTP

Remaining Setup Stores: 5

Comms Established

Collecting Temperature Data over a wide temperature range is a process detailed in the next flow diagram. This involves obtaining the detailed *Temp Code*, the *CorrDAC Code*, and if desired, the *VDDA Code* data needed for full correction implementation over the desired temperature range.

### **Step Environmental Chamber Through Temperature Range**

Move the environmental chamber through the desired temperature range and step size with the appropriate soak time. At each temperature point, collect the temperature data detailed in the Temperature Data Collection Flow section.

### ***Process Data Determine Correction Parameters***

Processing the temperature data is a procedure detailed separately in the Correction Implementation Flow sections. This involves evaluating the collected temperature data points and generating the information needed by the TMx00 for implementing the desired correction techniques.

### **Program the TMx00 & Validation**

Once the correction parameters have been determined from processing the temperature data, the TMx00 is programmed with the parameters for the correction technique that is to be used and validated with the resulting parameters. The correction technique and parameters can first be validated with the RAM based storage without having to send the data into the OTP storage. Select the desired correction technique, enter the correction values, and send the setup to the IC. The oscillator can be cycled over temperature and voltage as needed for testing. Correction values can be updated as many times as needed until the final values have been determined.

### ***Storing Values in the OTP Setup Block***

After the corrections are validated, the desired setup should be programmed into the TMx00 OTP as part of the oscillator setup. The values in the OTP will be used at each power-up once they are programmed. The TMx00 supports multiple OTP writes by providing 6 incremental storage blocks. If the contents of the OTP need to be changed, the new parameters are stored in the next storage block and the TMx00 increments a pointer to read from the new storage block upon startup. The process of writing an OTP storage block is defined in *TMx00 Programming Reference Manual* or can be accomplished through the TMx00 Control Software.

Hexius Semiconductor TMx00 Control Program

Options Help

**HEXIUS SEMICONDUCTOR**

General Setup Oscillator Setup **Correction Setup** THRM Calc Thermal Setup User Block

### Frequency Correction

**Correction Technique** ⓘ

☐ Disabled  
☐ Disabled with V/T Monitoring  
☒ Polynomial  
☐ Lookup Table

Correction DAC: 1000 0.000 V

**Correction Update Intervals** ⓘ

Temperature Sample Interval: 20 ms  
 Frequency Correction Interval: 30 ms (30 ms)  
 Averaging Intervals (Freq Corr): 10

**Temp Code Source** ⓘ

☒ From Internal Temp Sensor  
☐ From THRM

**Measure** ⓘ

Measure Temp, Volts & Correction

Measured IC Temp	26.8	C
IC Temp Code	2547	
THRM	2874	
Correction DAC Code	1000	
Correction DAC Volts	0	V
VDDA Code	2338	
VDDA Voltage	3.311	V

### Polynomial Correction

**Coefficients** ⓘ

Correction Order: 7

A0: 0  
 A1: 0  
 A2: 0  
 A3: 0  
 A4: 0  
 A5: 0  
 A6: 0  
 A7: 0  
 A8: 0  
 A9: 0  
 Mean: 0  
 Std Dev: 0

**Voltage Corr** ⓘ

V Corr Method: 1

B1: 0  
 B2: 0  
 B3: 0  
 C1: 0  
 C2: 0  
 C3: 0  
 B Mean: 0

### Lookup Table Correction

**Lookup Table** ⓘ

Temp Code	DAC Corr
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0

### Information

#### Correction Technique

'Disabled' disables any correction. The Correction DAC code (0-4095) is fed to the internal Varicap, independent of temperature.

'Disabled with V/T Monitoring' disables any correction and provides enhanced monitoring of the VDDA and IC Temperature Sensor/THRM input. This selection is used for initial characterization of the oscillator over temperature and voltage.

'Polynomial Correction' selects a polynomial curve fit correction technique. The internal Varicap or external varactor tuning voltage varies over temperature as set by the correction order coefficients, A0-A9.

'Table Lookup' selects a RAM based lookup table correction technique with up to 80 points. The internal Varicap or external varactor tuning voltage varies over temperature as set by the values stored in the lookup table.

IC Part Number: TM200

Firmware Rev: 1.8

**Connect, Restart Info** ⓘ

Connect to IC

Restart IC

Comms Established

(48) Transmitting data to device  
TX Data: 14 08 40 A4  
(48) Receiving data from device  
RX Data: 14 00 EC  
(49) Transmitting data to device  
TX Data: 15 EB  
(49) Receiving data from device  
RX Data: 15 22 09 08 B8  
(50) Transmitting data to device  
TX Data: 1F E1  
(50) Receiving data from device  
RX Data: 1F 3D F3 B6 46 C3 8A 2B 45 09

**Load or Save IC Setup** ⓘ

Load IC Setup from File

Save IC Setup to File

**Send Setup to IC (RAM based)** ⓘ

Send Current Setup Tab Items to IC

Send All Setup Tab Items to IC

**Load IC OTP Setup** ⓘ

Load Stored IC OTP Setup

**Store Setup into IC OTP** ⓘ

Store Current Setup into IC OTP

Remaining Setup Stores: 5

### 3 TEMPERATURE DATA COLLECTION FLOW

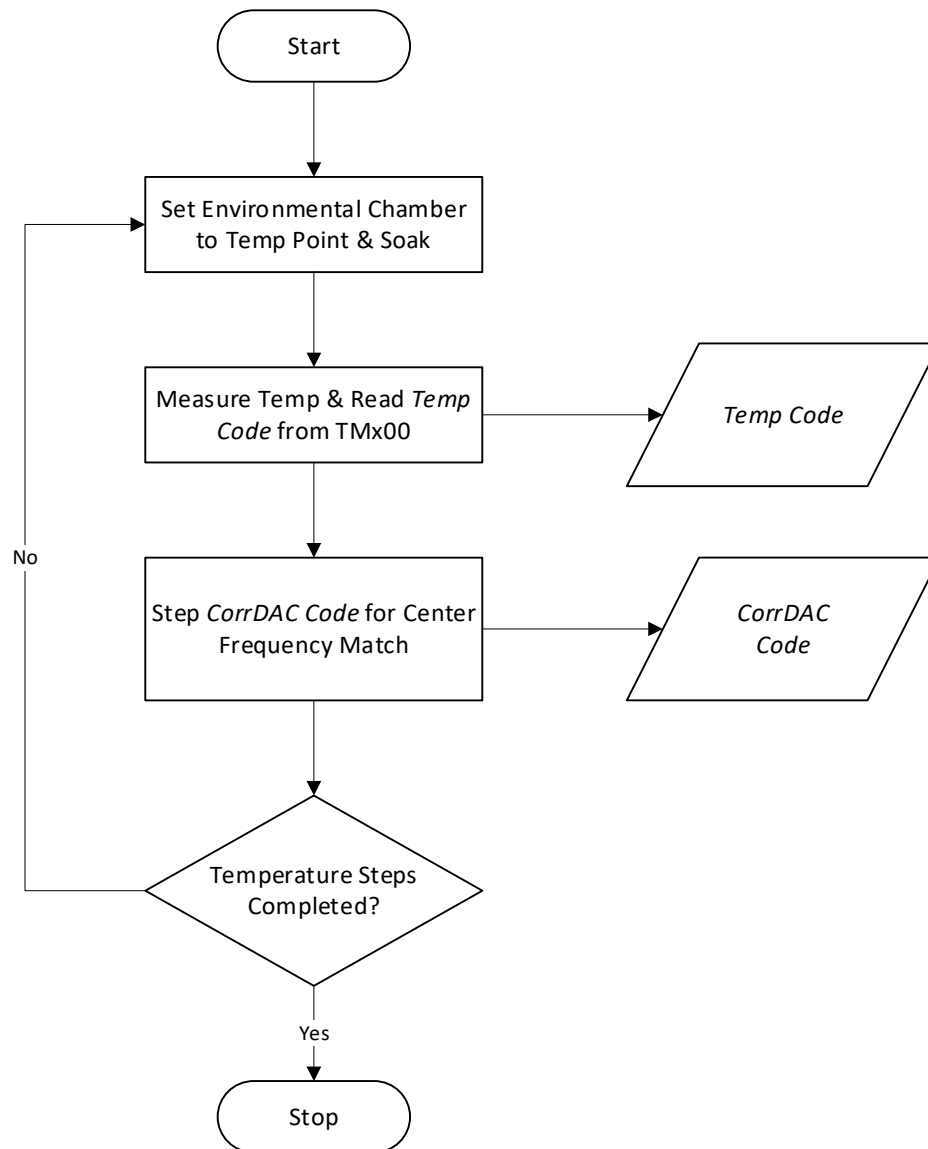


Figure 4 Temperature Data Collection Flow

#### Set Environmental Chamber to Starting Temperature Point

Set the environmental chamber to the desired starting point for the temperature sweep and soak as appropriate. The number of data points and step size in the temperature sweep will be dependent on the crystal characteristics and desired accuracy.

## Measure Temperature & Read Temp Code

The TMx00 Control Software can be used for *Temp Code* and *CorrDAC Code* data collection by manually measuring the *Temp Code* and entering the *CorrDAC Code* to determine the center frequency at each desired temperature point. An automated data collection process using I<sup>2</sup>C commands is a more efficient methodology.

Hexius Semiconductor TMx00 Control Program

Options Help

General Setup Oscillator Setup **Correction Setup** THRM Calc Thermal Setup User Block

**Frequency Correction**

**Correction Technique**

- ☒ Disabled
- ☐ Disabled with V/T Monitoring
- ☐ Polynomial
- ☐ Lookup Table

**Correction DAC** 1000 0.000 V

**Correction Update Intervals**

Temperature Sample Interval 20 ms

Frequency Correction Interval 30 ms (30 ms)

Averaging Intervals (Freq Corr) 10

**Temp Code Source**

- ☐ From Internal Temp Sensor
- ☐ From THRM

**Measure**

Measure Temp, Volts & Correction

Measured IC Temp 26.8 C

IC Temp Code 2547

THRM 2874

Correction DAC Code 1000

Correction DAC Volts 0 V

VDDA Code 2338

VDDA Voltage 3.311 V

**Polynomial Correction**

**Coefficients**

Correction Order 7

A0 0

A1 0

A2 0

A3 0

A4 0

A5 0

A6 0

A7 0

A8 0

A9 0

Mean 0

Std Dev 0

**Voltage Corr**

V Corr Method 1

B1 0

B2 0

B3 0

C1 0

C2 0

C3 0

B Mean 0

**Lookup Table Correction**

**Lookup Table**

Temp Code	DAC Corr
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0

**Information**

**Correction Technique**

'Disabled' disables any correction. The Correction DAC code (0-4095) is fed to the internal Varicap, independent of temperature.

'Disabled with V/T Monitoring' disables any correction and provides enhanced monitoring of the VDDA and IC Temperature Sensor/THRM input. This selection is used for initial characterization of the oscillator over temperature and voltage.

'Polynomial Correction' selects a polynomial curve fit correction technique. The internal Varicap or external varactor tuning voltage varies over temperature as set by the correction order coefficients, A0-A9.

'Table Lookup' selects a RAM based lookup table correction technique with up to 80 points. The internal Varicap or external varactor tuning voltage varies over temperature as set by the values stored in the lookup table.

IC Part Number: TM200

Firmware Rev: 1.8

**Connect, Restart Info**

Connect to IC

Restart IC

(48) Transmitting data to device  
TX Data: 14 08 40 A4  
(48) Receiving data from device  
RX Data: 14 00 EC  
(49) Transmitting data to device  
TX Data: 15 EB  
(49) Receiving data from device  
RX Data: 15 22 09 08 B8  
(50) Transmitting data to device  
TX Data: 1F E1  
(50) Receiving data from device  
RX Data: 1F 3D F3 B6 46 C3 8A 2B 45 09

Comms Established

**Load or Save IC Setup**

Load IC Setup from File

Save IC Setup to File

**Send Setup to IC (RAM based)**

Send Current Setup Tab Items to IC

Send All Setup Tab Items to IC

**Load IC OTP Setup**

Load Stored IC OTP Setup

**Store Setup into IC OTP**

Store Current Setup into IC OTP

Remaining Setup Stores 5

## I<sup>2</sup>C Command Option

Measure the temperature code, *Temp Code*, after the chamber is stabilized by issuing the following command. This example assumes a delay of 100.

Send: 0x14, 0x05, 0x64, 0x83

Recv: 0x14, 0x00, 0xEC

Read the data.

Send: 0x15, 0xEB

Recv: 0x15, data\_low, data\_high, 0x05, Checksum



### Determine the *CorrDAC Code* for Center Frequency

At each temperature point, the specific *CorrDAC Code* for the correct center frequency needs to be obtained. This is accomplished by stepping through the Correction DAC codes and measuring the resulting frequency until the appropriate *CorrDAC Code* is determined.

Step the Correction DAC code to set the oscillator to the desired center frequency through the TMx00 Control Software or by issuing the following I2C command:

Send: 0x03, 0x06, data\_low, data\_high, checksum

Recv: 0x03, 0x00, 0xFD

### Coarse Step the Correction DAC

Stepping through all the individual Correction DAC codes measuring the frequency is a slow process. A faster approach is to first perform a coarse step sweep of the Correction DAC codes, possibly 256 codes for each step. This will provide a coarse estimate of the correct *CorrDAC Code* to match the center frequency by linear interpolating the results.

### Fine Step the Correction DAC

Once an estimate of the *CorrDAC Code* is achieved, step the Correction DAC code by one count at a time around the estimated *CorrDAC Code* to determine the precise *CorrDAC Code*. A suggested range is  $\pm 64$  or 128 codes around the estimate.

Table 1 Example *Temp Code* & *CorrDAC Code* Raw Data Set

Temperature (C)	Temp Code	CorrDAC Code
-40	2005	2720
-30	2085	2318
-20	2167	2028
-10	2250	1856
0	2333	1794
10	2416	1822
20	2500	1906
25	2542	1961
30	2584	2014
40	2669	2124
50	2754	2164
60	2841	2142
70	2923	2015
80	3009	1723
90	3096	1032

## 4 VOLTAGE DATA COLLECTION FLOW

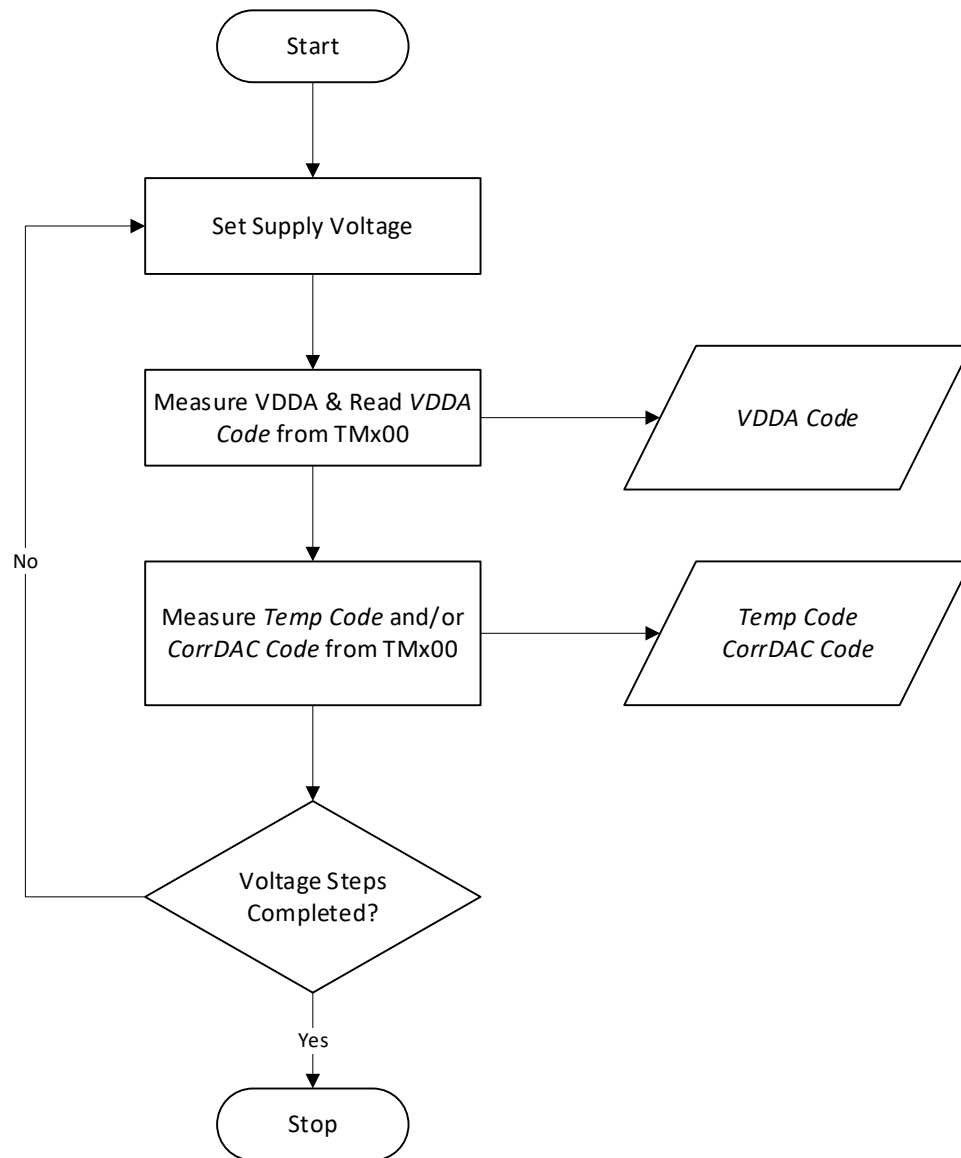


Figure 5 Voltage Data Collection Flow

### Set Supply Voltage

Set the supply voltage to the desired value.

### Measure Supply Voltage and Read VDDA Code

The TMx00 Control Software can be used for repeated for *VDDA Code* and *Temp Code* data collection by manually measuring the *VDDA Code* and *Temp Code* at each voltage step. An automated data collection process using I<sup>2</sup>C commands may be a more efficient methodology.

**Frequency Correction**

**Correction Technique**

- ☒ Disabled
- ☐ Disabled with V/T Monitoring
- ☐ Polynomial
- ☐ Lookup Table

Correction DAC: 1000 0.000 V

**Correction Update Intervals**

Temperature Sample Interval: 20 ms

Frequency Correction Interval: 30 ms (30 ms)

Averaging Intervals (Freq Corr): 10

**Temp Code Source**

- ☐ From Internal Temp Sensor
- ☐ From THRM

**Measure**

Measure Temp, Volts & Correction

Measured IC Temp: 25.0 °C

IC Temp Code: 2547

THRM: 2874

Correction DAC Code: 1000

Correction DAC Volts: 0 V

VDDA Code: 2338

VDDA Voltage: 3.311 V

**Polynomial Correction**

**Coefficients**

Correction Order: 7

A0: 0

A1: 0

A2: 0

A3: 0

A4: 0

A5: 0

A6: 0

A7: 0

A8: 0

A9: 0

Mean: 0

Std Dev: 0

**Voltage Corr**

V Corr Method: 1

B1: 0

B2: 0

B3: 0

C1: 0

C2: 0

C3: 0

B Mean: 0

**Lookup Table Correction**

**Lookup Table**

Temp Code	DAC Corr
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0

**Information**

**Correction Technique**

'Disabled' disables any correction. The Correction DAC code (0-4095) is fed to the internal Varicap, independent of temperature.

'Disabled with V/T Monitoring' disables any correction and provides enhanced monitoring of the VDDA and IC Temperature Sensor/THRM input. This selection is used for initial characterization of the oscillator over temperature and voltage.

'Polynomial Correction' selects a polynomial curve fit correction technique. The internal Varicap or external varactor tuning voltage varies over temperature as set by the correction order coefficients, A0-A9.

'Table Lookup' selects a RAM based lookup table correction technique with up to 80 points. The internal Varicap or external varactor tuning voltage varies over temperature as set by the values stored in the lookup table.

IC Part Number: TM200

Firmware Rev: 1.8

**Connect, Restart Info**

Connect to IC

Restart IC

(48) Transmitting data to device  
TX Data: 14 08 40 A4  
(48) Receiving data from device  
RX Data: 14 00 EC  
(49) Transmitting data to device  
TX Data: 15 EB  
(49) Receiving data from device  
RX Data: 15 22 09 08 B8  
(50) Transmitting data to device  
TX Data: 1F E1  
(50) Receiving data from device  
RX Data: 1F 3D F3 B6 46 C3 8A 2B 45 09

**Load or Save IC Setup**

Load IC Setup from File

Save IC Setup to File

**Send Setup to IC (RAM based)**

Send Current Setup Tab Items to IC

Send All Setup Tab Items to IC

**Load IC OTP Setup**

Load Stored IC OTP Setup

**Store Setup into IC OTP**

Store Current Setup into IC OTP

Remaining Setup Stores: 5

Comms Established

## I<sup>2</sup>C Command Option

Measure the voltage code, *VDDA Code*, after the supply voltage is set to the desired value by issuing the following command. This example assumes a delay of 100.

Send: 0x14, 0x08, 0x64, 0x80

Recv: 0x14, 0x00, 0xEC

Read the data.

Send: 0x15, 0xEB

Recv: 0x15, data\_low, data\_high, 0x05, Checksum

## Measure Temperature & Read *Temp Code*

At each supply voltage point, read the *Temp Code* to measure the shift.

### I<sup>2</sup>C Command Option

Measure the temperature code, *Temp Code*, after the chamber is stabilized by issuing the following command. This example assumes a delay of 100.

Send: 0x14, 0x05, 0x64, 0x83

Recv: 0x14, 0x00, 0xEC

Read the data.

Send: 0x15, 0xEB

Recv: 0x15, data\_low, data\_high, 0x05, Checksum

Table 2 Example *VDDA Code* & *Temp Code* Raw Data Set

<i>Supply (V)</i>	<i>VDDA Code</i>	<i>Temp Code</i>
2.95	2083	2316
3.00	2118	2318
3.05	2153	2320
3.10	2189	2322
3.15	2224	2324
3.20	2259	2326
3.25	2295	2328
3.30	2330	2330
3.35	2365	2332
3.40	2401	2334
3.45	2436	2336
3.50	2471	2338
3.55	2506	2340
3.60	2542	2342
3.65	2577	2344

## 5 LOOKUP TABLE IMPLEMENTATION FLOW

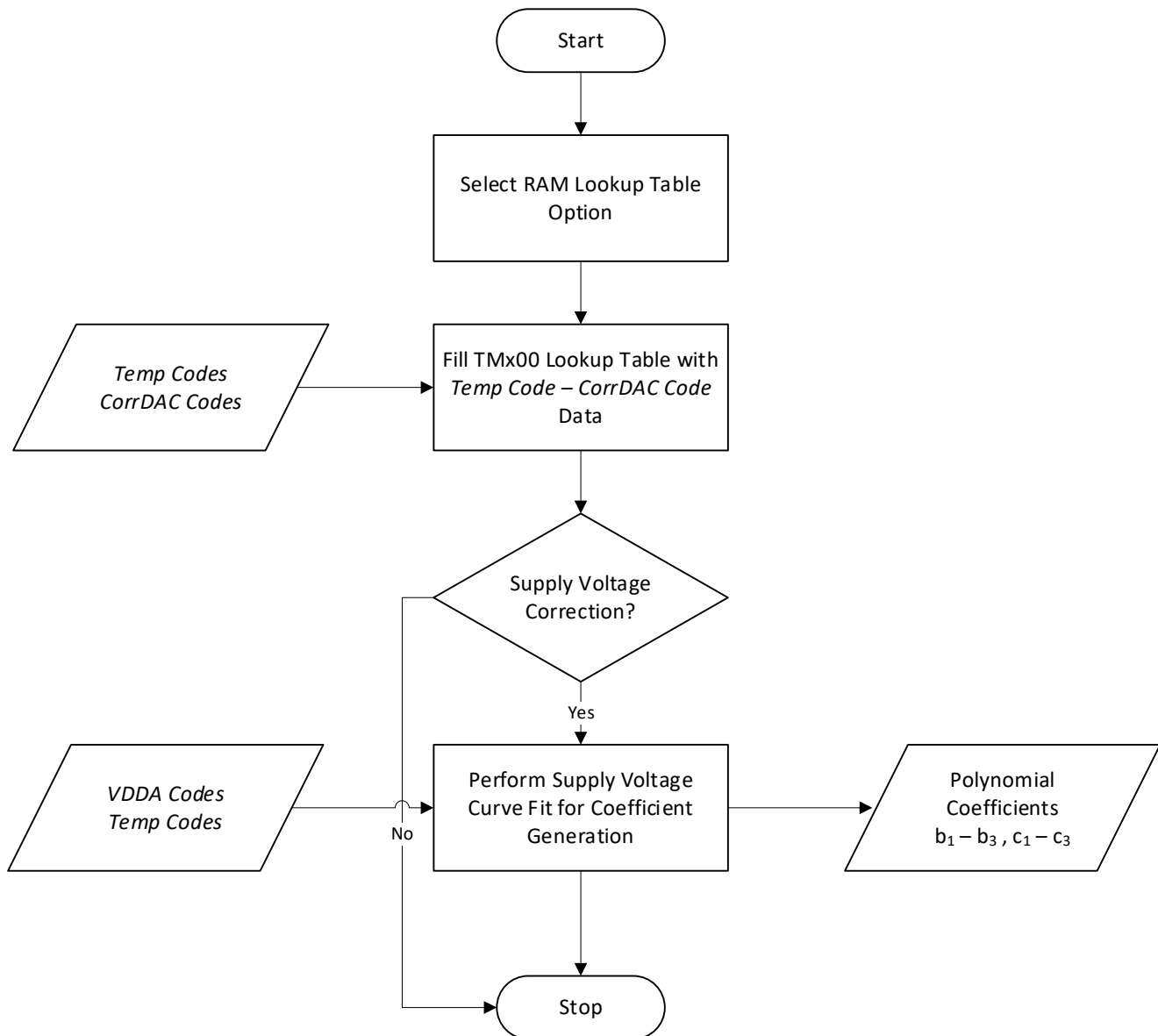


Figure 6 Lookup Table Implementation Flow

The Linear Lookup Table correction technique uses the measured reading from the IC temperature sensor or external thermistor (*Temp Code*) and generates it into a *CorrDAC Code* via a lookup table.

The lookup table is arranged from the lowest *Temp Code* to the highest *Temp Code* with up to 80 defined points. The spacing between the points is user defined to account for changing crystal temperature coefficient slopes. The appropriate DAC correction code (*CorrDAC Code*) between the defined *Temp Code* points is calculated with a linear interpolation technique

## Select Lookup Table

The Lookup Table correction algorithm in the TMx00 uses up to 80 data points of *Temp Code* – *CorrDAC* Codes pairings.

Hexius Semiconductor TMx00 Control Program

Options Help

**HEXIUS SEMICONDUCTOR**

General Setup Oscillator Setup **Correction Setup** THRM Calc Thermal Setup User Block

### Frequency Correction

**Correction Technique**

☐ Disabled  
☐ Disabled with V/T Monitoring  
☐ Polynomial  
☒ **Lookup Table**

Correction DAC: 1000 0.000 V

**Correction Update Intervals**

Temperature Sample Interval: 20 ms  
 Frequency Correction Interval: 30 ms (30 ms)  
 Averaging Intervals (Freq Corr): 10

**Temp Code Source**

☒ From Internal Temp Sensor  
☐ From THRM

**Measure**

Measure Temp, Volts & Correction	
Measured IC Temp	26.8 C
IC Temp Code	2547
THRM	2874
Correction DAC Code	1000
Correction DAC Volts	0 V
VDDA Code	2338
VDDA Voltage	3.311 V

### Polynomial Correction

**Coefficients**

Correction Order: 7

A0: 0 A1: 0 A2: 0 A3: 0 A4: 0 A5: 0 A6: 0 A7: 0 A8: 0 A9: 0

Mean: 0 Std Dev: 0

**Voltage Corr**

V Corr Method: 1

B1: 0 B2: 0 B3: 0 C1: 0 C2: 0 C3: 0 B Mean: 0

### Lookup Table Correction

**Lookup Table**

Temp Code	DAC Corr
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0

### Information

#### Lookup Table

The Lookup Table Correction technique uses the measured temperature reading (Temp Code) and translates it into a Correction DAC code (CorrDAC) via a lookup table with linear interpolation. The DAC Corr value is an integer with the range of 0 to 4095 and has a corresponding voltage.

The Lookup Table is arranged from the lowest temperature to the highest temperature with up to 80 points.

Temperature readings between defined points are linearly interpolated and converted in the appropriate Correction DAC code.

Temperature codes below the lowest table entry generate the lowest Correction DAC code. Temperature codes above the highest table entry generate the highest Correction DAC code.

Zero values for Temp Code and CorrDAC mark the end of the table if less than 80 elements are filled.

Voltage correction can also be used for enhanced accuracy by adjusting the lookup calculations.

IC Part Number: TM200  
Firmware Rev: 1.8

**Connect, Restart Info**

Connect to IC  
Restart IC

(48) Transmitting data to device  
TX Data: 14 08 40 A4  
(48) Receiving data from device  
RX Data: 14 00 EC  
(49) Transmitting data to device  
TX Data: 15 EB  
(49) Receiving data from device  
RX Data: 15 22 09 08 B8  
(50) Transmitting data to device  
TX Data: 1F E1  
(50) Receiving data from device  
RX Data: 1F 3D F3 B6 46 C3 8A 2B 45 09

**Load or Save IC Setup**

Load IC Setup from File  
Save IC Setup to File

**Send Setup to IC (RAM based)**

Send Current Setup Tab Items to IC  
Send All Setup Tab Items to IC

**Load IC OTP Setup**

Load Stored IC OTP Setup

**Store Setup into IC OTP**

Store Current Setup into IC OTP

Remaining Setup Stores: 5

Comms Established

## Fill TMx100 Lookup Table

Fill in the Lookup Table via the TMx00 Control Software by taking the *Temp Code* and *CorrDAC Code* value gathered during the temperature data collection (Table 1) and entering it into the entry table. The lookup table data needs to be arranged from the lowest *Temp Code* to the highest *Temp Code*. The end of the table occurs when either 1) both the *Temp Code* and the *CorrDAC Code* values for an entry are zero, or 2) all table entries are filled.

If the measured *Temp Code* value is below the minimum value in the table, the *CorrDAC Code* value is set to the value that corresponds to the lowest *Temp Code* in the table. If the *Temp Code* is above the maximum value in the table, the *CorrDAC Code* value is set to the value that corresponds to the highest *Temp Code* in the table.

If the measured temperature is the exact value of a *Temp Code*, the corresponding *CorrDAC Code* value is used for the correction. For measured temperatures between the Lookup Table's *Temp Codes*, the *CorrDAC Code* values are linearly interpolated. Since the temperature coefficient curve of a crystal is non-linear, the amount and spacing of the *Temp Code* - *CorrDAC Code* entries should be optimized to minimize the correction error (Figure 7) depending on the rate of change of the temperature coefficient curve within a certain temperature range. It should be noted that the *Temp Code* values in the Lookup Table do not need to be equally spaced.

The screenshot displays the 'Hexius Semiconductor TMx00 Control Program' window. The 'Correction Setup' tab is active, showing three sub-sections: Frequency Correction, Polynomial Correction, and Lookup Table Correction. The 'Lookup Table Correction' section is highlighted with a red box, showing a table with 17 entries. The table has two columns: 'Temp Code' and 'DAC Corr'. The values are as follows:

Temp Code	DAC Corr
1986	2512
2002	2426
2019	2344
2035	2271
2052	2203
2069	2143
2086	2097
2103	2047
2120	2008
2136	1977
2154	1953
2170	1935
2187	1925
2204	1321
2220	1923
2237	1931
2254	1945

Below the table, there are buttons for 'Load IC Setup from File' and 'Save IC Setup to File'. To the right of the table, there is an 'Information' section titled 'Lookup Table' which explains the correction technique and provides additional details about the table's structure and usage.

At the bottom of the window, there is a status bar showing 'Comms Established' and a log of data transmissions and receptions.

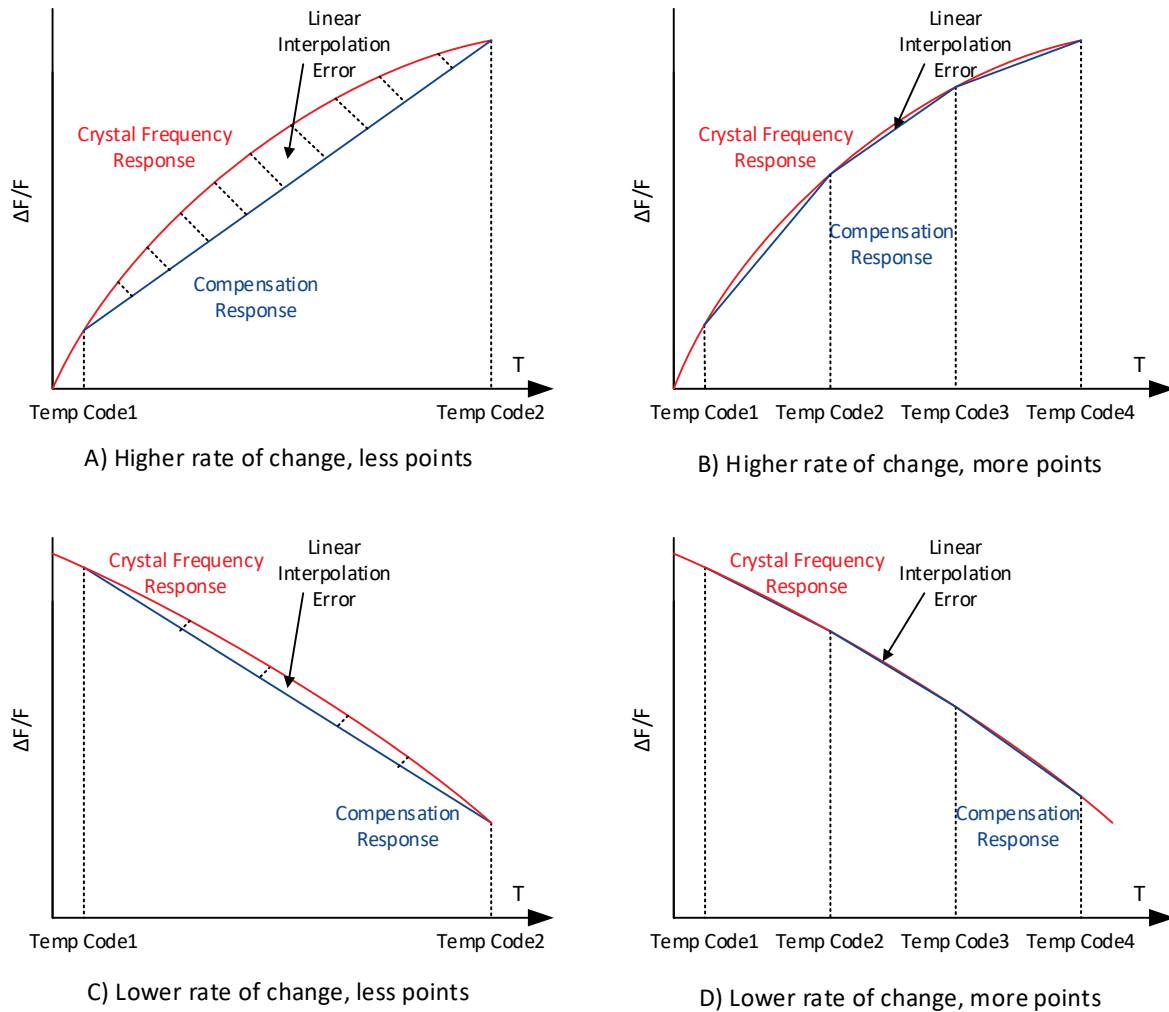


Figure 7 Linear Interpolation Temp Code Spacing Considerations

### I<sup>2</sup>C Command Option

The *lookup table* variable contains up to 80 pairs of temperature code readings for the table lookup and tuning DAC code values, 3 bytes per pair.

The first 1.5 bytes in each entry contain the *Temp Code*, and the last 1.5 bytes contain the *CorrDAC Code* value. The most significant bit of the *Temp Code* is in the most significant bit of the first byte. The most significant bit of the *CorrDAC Code* starts at the most significant bit of the bottom nibble in the middle byte. The *Temp Code* readings must be monotonic from low to high. The *CorrDAC Code* values do not have that restriction.

Please reference the *TMx00 Programming Reference Manual* for more information.

### Perform Supply Voltage Correction (Optional)

Using the Lookup Table for temperature correction instead of the polynomial correction does not limit the user from the Supply Voltage Correction. The Supply Voltage Correction flow is described in section 7.



## 6 TEMPERATURE POLYNOMIAL IMPLEMENTATION FLOW

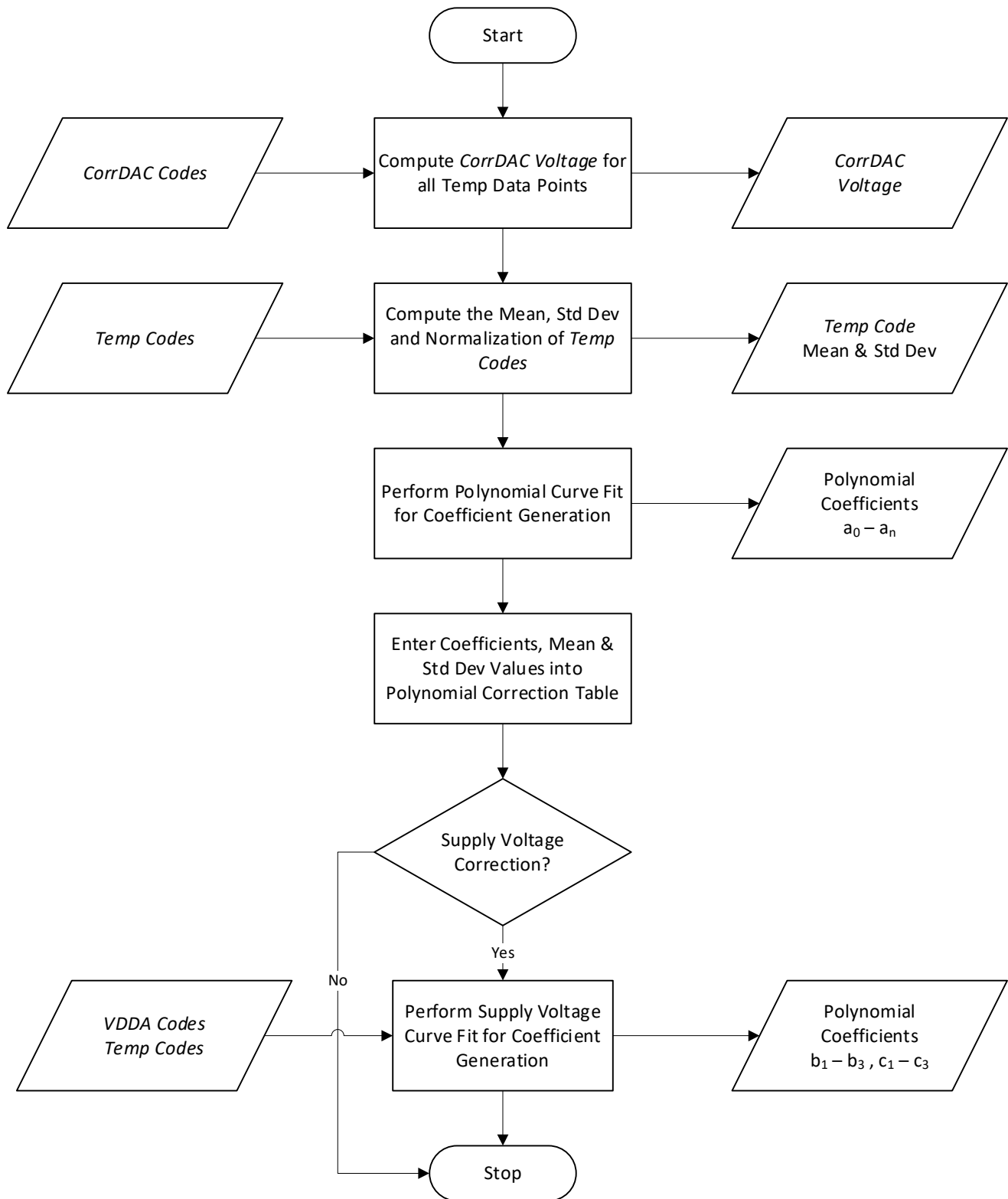


Figure 8 Temperature Polynomial Implementation Flow

### Compute the *CorrDAC Voltages* for all Temperature Points

Using the *Temp Code* and *CorrDAC Code* raw data, calculate the *CorrDAC Voltage* for each temperature point. The polynomial curve fit calculation requires the *CorrDAC Voltage* and not the *CorrDAC Code* (Lookup Table technique).

The *CorrDAC Voltage* is the resulting Correction DAC output voltage for a given *CorrDAC Code*. It is calculated by multiplying the Correction DAC's reference voltage by the ratio of the *CorrDAC Code* to the Correction DAC's full-scale code (4095).

$$\text{CorrDAC Voltage} = \frac{\text{CorrDAC Code}}{4095} * \text{BYPASS pin Voltage}$$

Example:

Table 3 CorrDAC Voltage Calculation Example

Temperature (C)	Temp Code	CorrDAC Code	CorrDAC Voltage
-40	2005	2720	1.92625153
-30	2085	2318	1.64156288
-20	2167	2028	1.43619048
-10	2250	1856	1.31438339
0	2333	1794	1.27047619
10	2416	1822	1.29030525
20	2500	1906	1.34979243
25	2542	1961	1.38874237
30	2584	2014	1.42627595
40	2669	2124	1.50417582
50	2754	2164	1.53250305
60	2841	2142	1.51692308
70	2923	2015	1.42698413
80	3009	1723	1.22019536
90	3096	1032	0.73084249

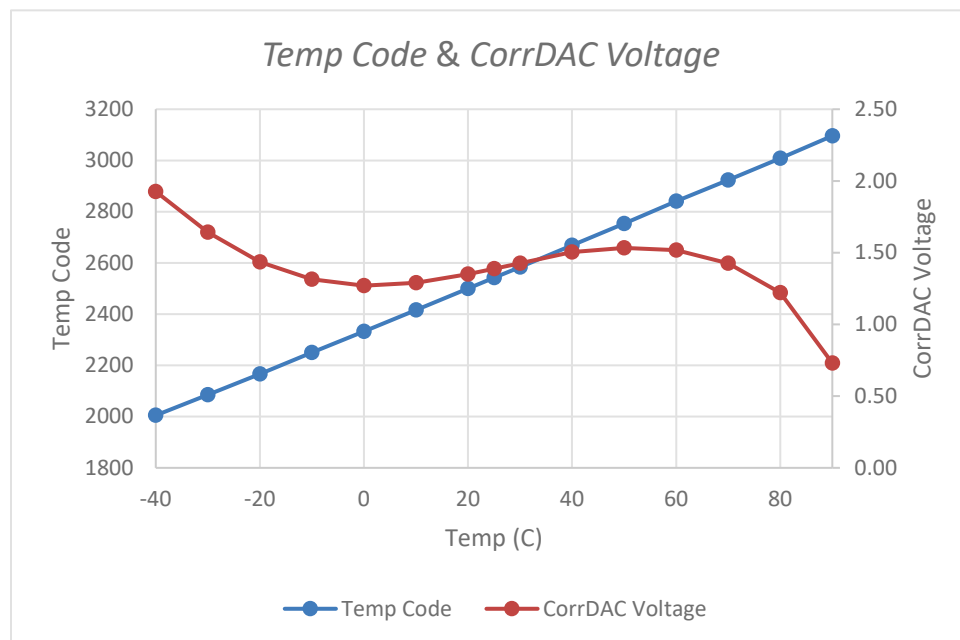


Figure 9 Temp Code & CorrDAC Voltage Over Temperature Sweep

### Compute the Mean, Std Dev, and Normalization of *Temp Codes*

Processing the *Temp Code* data for mean and standard deviation may enhance the accuracy of the correction equations. The result is the *Normalized Temp Code*.

Using the Table 3 *Temp Code* values, the mean is:

$$\begin{aligned} \text{sample}_i &= \text{each sample} \\ N &= \text{number of samples} \\ \text{Mean} &= \frac{\Sigma(\text{sample}_i)}{N} = 2544.9333 \end{aligned}$$

Using the Table 3 *Temp Code* values, the standard deviation is

$$\begin{aligned} \sigma &= \sqrt{\frac{\Sigma(\text{sample}_i - \text{mean})^2}{N}} \\ \sigma &= 327.2820 \end{aligned}$$

Normalized *Temp Code* values are computed by:

$$\text{Normalized Temp Code} = \frac{\text{Temperature Code} - \text{Mean}}{\sigma}$$

Table 4 Example with Normalized Temperature Code

Temperature (C)	Temp Code	Normalized Temp Code	CorrDAC Code	CorrDAC Voltage
-40	2005	-1.649749389	2720	1.92625153
-30	2085	-1.405311895	2318	1.64156288
-20	2167	-1.154763463	2028	1.43619048
-10	2250	-0.901159562	1856	1.31438339
0	2333	-0.647555662	1794	1.27047619
10	2416	-0.393951762	1822	1.29030525
20	2500	-0.137292393	1906	1.34979243
25	2542	-0.008962708	1961	1.38874237
30	2584	0.119366976	2014	1.42627595
40	2669	0.379081814	2124	1.50417582
50	2754	0.638796652	2164	1.53250305
60	2841	0.904622427	2142	1.51692308
70	2923	1.155170859	2015	1.42698413
80	3009	1.417941165	1723	1.22019536
90	3096	1.68376694	1032	0.73084249
Mean	2544.9333			
Std Dev	327.2820			

## Perform Temperature Correction Polynomial Curve Fit for Coefficient Generation

The general form of the polynomial correction equation is:

$$Corr_{temp}(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7 + a_8x^8 + a_9x^9$$

$x$  = *Normalize Temp Code: TempCode normalized for Mean and Standard Deviation*

$Corr_{temp}(x)$  = *Correction DAC output voltage*

Assuming 7<sup>th</sup> order correction, the equation is:

$$Corr_{temp}(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7$$

The term  $x$  is the *Normalized Temp Code*, and  $Corr_{temp}(x)$  is the Correction DAC voltage output.

Table 5 Data Values Entered Into Polynomial Curve Fit Calculation

$x$	$Corr_{temp}(x)$
-1.649749389	1.92625153
-1.405311895	1.64156288
-1.154763463	1.43619048
-0.901159562	1.31438339
-0.647555662	1.27047619
-0.393951762	1.29030525
-0.137292393	1.34979243
-0.008962708	1.38874237
0.119366976	1.42627595
0.379081814	1.50417582
0.638796652	1.53250305
0.904622427	1.51692308
1.155170859	1.42698413
1.417941165	1.22019536
1.68376694	0.73084249

If the *Normalized Temp Code* and *CorrDAC Voltage* values are inputted into a polynomial curve fit calculator (MATLAB, Stats.blue, etc), the following coefficients are generated:

Table 6 Temperature Correction Polynomial Coefficients

Coefficient	Estimate	Standard Error
$a_0$	1.3932454	0.00185
$a_1$	0.3157902	0.00672
$a_2$	0.0033945	0.00818
$a_3$	-0.2853885	0.0168
$a_4$	0.0445517	0.00771
$a_5$	0.0517127	0.01191
$a_6$	-0.0178741	0.00186
$a_7$	-0.0128117	0.00245

The polynomial correction equation then becomes:

$$Corr_{temp}(x) = 1.3933 + 0.3158x + 0.0034x^2 - 0.2854x^3 + 0.0446x^4 + 0.0517x^5 - 0.0179x^6 - 0.0128x^7$$

The numeric values shown in this example are illustrated with limited digits of precision. All significant digits should be maintained through the curve fitting process.

The TMx00 firmware automatically converts the DAC voltage computed by the equation into the needed DAC code for setting the desired frequency.

### Enter Coefficients, Mean, & Standard Deviation Values

Select the Polynomial Correction technique and set the Correction Order. Enter the coefficient values  $a_0$ – $a_9$ , mean, and standard deviation calculated into the TMx00 Control Software Polynomial Correction Table.

**Frequency Correction**

**Correction Technique**

- ☐ Disabled
- ☐ Disabled with V/T Monitoring
- ☒ Polynomial
- ☐ Lookup Table

Correction DAC: 1000 0.000 V

**Correction Update Intervals**

Temperature Sample Interval: 20 ms

Frequency Correction Interval: 30 ms (30 ms)

Averaging Intervals (Freq Corr): 10

**Temp Code Source**

- ☒ From Internal Temp Sensor
- ☐ From THRM

**Measure**

Measure Temp, Volts & Correction

Measured IC Temp	26.8	C
IC Temp Code	2547	
THRM	2874	
Correction DAC Code	1000	
Correction DAC Volts	0	V
VDDA Code	2338	
VDDA Voltage	3.311	V

**Polynomial Correction**

**Coefficients**

Correction Order: 7

A0	1.3933
A1	0.3158
A2	0.0034
A3	-0.2854
A4	0.0446
A5	0.0517
A6	-0.0179
A7	-0.0128
A8	
A9	
Mean	2544.9333
Std Dev	327.2820

**Voltage Corr**

V Corr Method: 1

B1	0
B2	0
B3	0
C1	0
C2	0
C3	0
B Mean	0

**Lookup Table Correction**

**Lookup Table**

Temp Code	DAC Corr
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0

**Information**

### Correction Coefficients

The Polynomial Correction Technique uses the measured temperature reading (Temp Code) and calculates a Correction DAC voltage via polynomial curve fit using the correction order coefficients,  $a_0$ – $a_9$ .

The correction process computes the DAC voltage (0 to 2.9 V) needed for best correction. The correction value is converted into a 12-bit DAC input digital code, 0 for 0V, and 4095 for 2.9V

Obtaining the correction coefficient values is a separate process and the derived values are entered as floating point numbers.

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7 + a_8x^8 + a_9x^9$$

Correction Order defines the polynomial curve fit correction order, 1 through 9.

$x$  = Temp Code normalized to Mean and Standard Deviation over the Temp Code Range.

$f(x)$  = Correction DAC voltage to Varicap

Mean is the average value and Std Dev is the standard deviation of the Temperature Code from the curve fit process.

Best curve fit results are achieved when the coefficients values are entered with as many digits of precision as practical.

IC Part Number: TM200

Firmware Rev: 1.8

**Connect, Restart Info**

Connect to IC

Restart IC

Comms Established

(48) Transmitting data to device  
TX Data: 14 08 40 A4  
(48) Receiving data from device  
RX Data: 14 00 EC  
(49) Transmitting data to device  
TX Data: 15 EB  
(49) Receiving data from device  
RX Data: 15 22 09 08 B8  
(50) Transmitting data to device  
TX Data: 1F E1  
(50) Receiving data from device  
RX Data: 1F 3D F3 B6 46 C3 8A 2B 45 09

**Load or Save IC Setup**

Load IC Setup from File

Save IC Setup to File

**Send Setup to IC (RAM based)**

Send Current Setup Tab Items to IC

Send All Setup Tab Items to IC

**Load IC OTP Setup**

Load Stored IC OTP Setup

**Store Setup into IC OTP**

Store Current Setup into IC OTP

Remaining Setup Stores: 5

### I<sup>2</sup>C Command Option

The resultant terms are stored in the setup block, each as a single precision floating point value.  $a_0$  is stored in `corr_a0`,  $a_1$  is stored in `corr_a1`, up through the correction order specified. The mean (2544.9333) is stored in `corr_mean`, and the standard deviation (327.2820) is stored in `corr_std`.

## 7 SUPPLY VOLTAGE IMPLEMENTATION FLOW

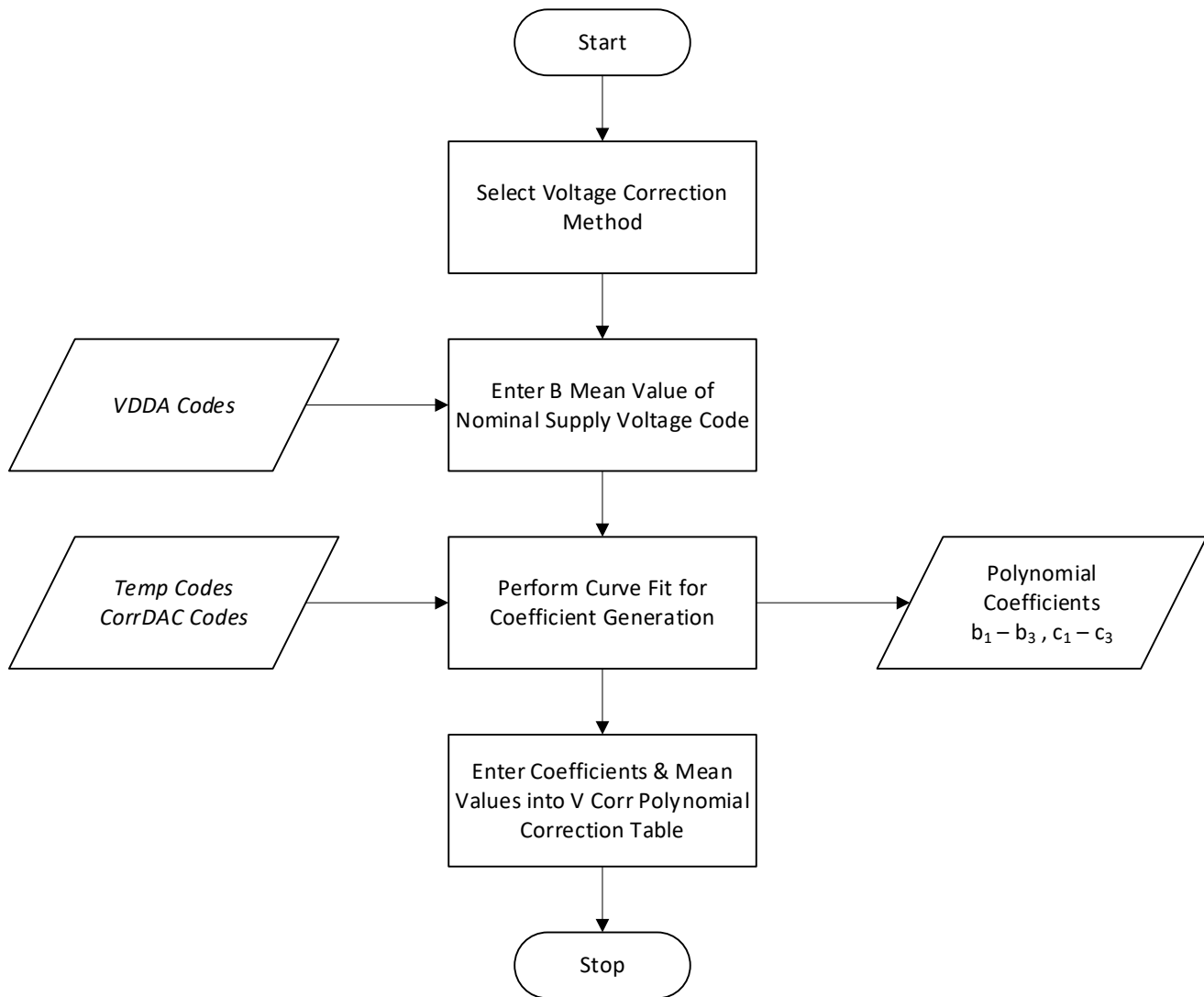


Figure 10 Supply Voltage Polynomial Implementation Flow

The *CorrDAC Voltage* values are not overly sensitive to supply voltage (VDDA) variation, but the *Temp Code Normalized* values have a small dependency on supply voltage. This means that supply voltage variations will shift the *Temp Code* values. The supply voltage polynomial curve fit correction techniques are optional enhanced correction methods that generate adjustment terms to account for this shift.

A  $Corr_{voltage}$  term corrects for frequency shifts over supply voltage variation and uses the coefficients,  $b_1 - b_3$ . (Most common since the supply voltage variations will shift the *Temp Code* values with a linear relationship.)

A  $Corr_{voltage\_temp}$  term corrects for frequency shifts from cross correlated supply voltage and temperature variation and uses the coefficients,  $c_1 - c_3$ . (Applicable to very high stability requirements)

The user selects whether these correction terms are applied to either the *Temp Code* or *CorrDAC* Code depending on the architecture and sensitivity of supply voltage movement of the module. *Temp Code* or *CorrDAC* adjustments each have further options regarding the order of correction desired.

### Temp Code Adjustment Option

Applying the correction by adjusting the *Temp Code* is defined as

$$Temp_{ADJ} = TempCodeNormalized + \Delta TempCode(\Delta VDDA)$$

Where:

$Temp_{ADJ}$  is the new value input into the main  $Corr_{temp}(x)$  function for frequency correction

$\Delta TempCode$  is the *TempCode* adjustment based on voltage and temperature measurements

$\Delta VDDA$  is the *VDDA Code* normalized to the Mean and Standard Deviation

#### Temp Code $Corr_{voltage}$ 1<sup>st</sup> Order Correction Option – MOST COMMON

$$\Delta TempCode(\Delta VDDA) = Corr_{voltage}(\Delta VDDA) = b_1(\Delta VDDA)$$

#### Temp Code $Corr_{voltage}$ 3<sup>rd</sup> Order Correction Option

$$\Delta TempCode(\Delta VDDA) = Corr_{voltage}(\Delta VDDA) = b_1(\Delta VDDA) + b_2(\Delta VDDA)^2 + b_3(\Delta VDDA)^3$$

#### Temp Code $Corr_{voltage}$ & $Corr_{voltage\_temp}$ 3<sup>rd</sup> Order Correction Option

$$\begin{aligned} \Delta TempCode(\Delta VDDA) &= Corr_{voltage}(\Delta VDDA) + Corr_{voltage\_temp}(\Delta VDDA) \\ &= b_1(\Delta VDDA) + b_2(\Delta VDDA)^2 + b_3(\Delta VDDA)^3 + c_1(\Delta VDDA)(TempCode) \\ &\quad + c_2(\Delta VDDA)^2(TempCode) + c_3(\Delta VDDA)(TempCode)^2 \end{aligned}$$

### CorrDAC Code Adjustment Option

Applying the correction by adjusting the *CorrDAC Code* is defined as

$$CorrDAC_{ADJ} Code = CorrDAC Code + \Delta CorrDAC Code(\Delta VDDA) \text{ (when using the Lookup Table Correction)}$$

Or

$$CorrDAC_{ADJ} Voltage = CorrDAC Voltage + \Delta CorrDAC Voltage(\Delta VDDA) \text{ (when using the Polynomial Correction)}$$

Where:

$CorrDAC_{ADJ} Code$  is the new *CorrDAC Code* value output for frequency correction (Lookup Table Correction)

$CorrDAC_{ADJ} Voltage$  is the new *CorrDAC Voltage* output for frequency correction (Polynomial Correction)

$\Delta CorrDAC$  is the *CorrDAC Code/Voltage* adjustment based on voltage and temperature measurements.

$\Delta VDDA$  is the *VDDA Code* normalized to the Mean and Standard Deviation.

#### CorrDAC Code $Corr_{voltage}$ 1<sup>st</sup> Order Correction Option - MOST COMMON

$$\Delta CorrDAC Code(\Delta VDDA) = Corr_{voltage}(\Delta VDDA) = b_1(\Delta VDDA)$$

#### CorrDAC Code $Corr_{voltage}$ 3<sup>rd</sup> Order Correction Option

$$\Delta CorrDAC Code(\Delta VDDA) = Corr_{voltage}(\Delta VDDA) = b_1(\Delta VDDA) + b_2(\Delta VDDA)^2 + b_3(\Delta VDDA)^3$$

#### CorrDAC Code $Corr_{voltage}$ & $Corr_{voltage\_temp}$ 3<sup>rd</sup> Order Correction Option

$$\begin{aligned} \Delta CorrDAC Code(\Delta VDDA) &= Corr_{voltage}(\Delta VDDA) + Corr_{voltage\_temp}(\Delta VDDA) \\ &= b_1(\Delta VDDA) + b_2(\Delta VDDA)^2 + b_3(\Delta VDDA)^3 + c_1(\Delta VDDA)(TempCode) \\ &\quad + c_2(\Delta VDDA)^2(TempCode) + c_3(\Delta VDDA)(TempCode)^2 \end{aligned}$$

The measured VDDA voltage is one-half of the VDDA value, or a nominal code of 2330 (0x091A) for a 3.3V VDDA value. The VDDA voltage is varied across the supply range, and the resulting temperature code is measured at a single temperature value.

### Temp Code Adjustment Example - Linear Correction

The data below is from a supply voltage sweep that collects the *VDDA Code* and *Temp Code* information needed for a correction implementation. This example assumes the  $\Delta$ frequency over  $\Delta$ VDDA is only caused by *Temp Code* shifts.

Table 7 Voltage Correction Temp Code Data Set Example

Supply (V)	VDDA Code	Temp Code
3.135	2213	2542
3.2175	2272	2544
3.3	2330	2546
3.3825	2388	2548
3.465	2446	2550

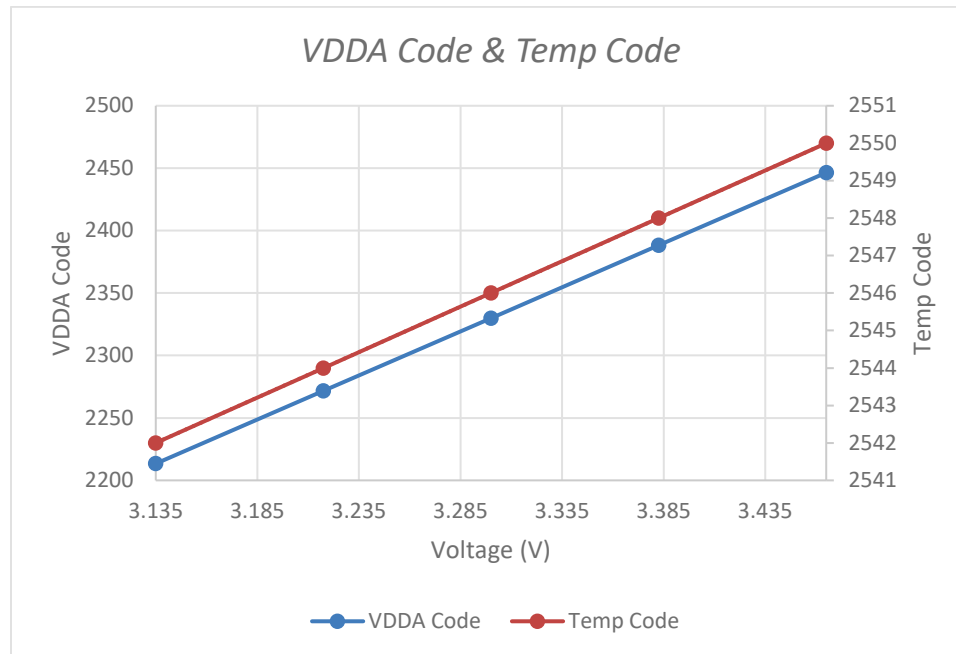


Figure 11 VDDA Code & Temp Voltage Over Supply Voltage Sweep

The relationship between *VDDA Code* and *Temp Code* is linear. Therefore, select V Corr Method 1 which is:

#### Temp Code Corr<sub>voltage</sub> 1<sup>st</sup> Order Correction Option

$$\Delta \text{TempCode} (\Delta \text{VDDA}) = \text{Corr}_{\text{voltage}} (\Delta \text{VDDA}) = b_1 (\Delta \text{VDDA})$$

For this correction technique, the B Mean and B1 terms need to be entered.



**Frequency Correction**

Correction Technique: ☐ Disabled, ☐ Disabled with V/T Monitoring, ☒ Polynomial, ☐ Lookup Table

Correction DAC: 1000, 0.000 V

Correction Update Intervals: Temperature Sample Interval: 20 ms, Frequency Correction Interval: 30 ms (30 ms), Averaging Intervals (Freq Corr): 10

Temp Code Source: ☒ From Internal Temp Sensor, ☐ From THRM

**Measure**

Measure Temp, Volts & Correction	
Measured IC Temp	26.9 C
IC Temp Code	2548
THRM	2827
Correction DAC Code	1000
Correction DAC Volts	0 V
VDDA Code	2338
VDDA Voltage	3.311 V

**Polynomial Correction**

Coefficients: Correction Order: 7

A0: 1.3933, A1: 0.3158, A2: 0.0034, A3: -0.2854, A4: 0.0446, A5: 0.0517, A6: -0.0179, A7: -0.0128, A8: , A9: , Mean: 2544.9333, Std Dev: 327.2820

**Voltage Corr**

V Corr Method: 1

B1: 0, B2: 0, B3: 0, C1: 0, C2: 0, C3: 0, B Mean:

**Lookup Table Correction**

Temp Code	DAC Corr
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0

**Information**

**Voltage Correction**

V Corr Method defines an optional enhanced correction algorithm that generates an adjustment term, compensating for variations over supply voltage (VDDA) or supply voltage over temperature. Several different correction methods are available, defined by the value of V Corr Order.

$\Delta VDDA$  = VDDA normalized to Mean and Standard Deviation. Mean (B\_Mean) is the VDDA Code for 3.3V applied to the VDDA pin. The Standard Deviation is computed internally to equal  $\pm 3 (3\sigma)$  when the supply is 3.3V  $\pm 5\%$ .

temp\_code = The value from the Temp Code Source (Internal IC Sensor or THRM).

**TempCode Adjustment Option**

$\Delta VDDA$  = VDDA normalized to Mean and Standard Deviation. Mean (B\_Mean) is the VDDA Code for 3.3V applied to the VDDA pin. The Standard Deviation is computed internally to equal  $\pm 3 (3\sigma)$  when the supply is 3.3V  $\pm 5\%$ .

temp\_code = The value from the Temp Code Source (Internal IC Sensor or THRM).

$\Delta TempCode = TempCodeNormalized + Temp_{Adj}(\Delta VDDA)$

$\Delta TempCode$  is the new TempCodeNormalized value input into the main  $Corr_{Temp}(x)$  function for frequency correction.

**V Corr Method = 0**

No correction applied.

**V Corr Method = 1, TempCode = Corr<sub>Voltage</sub> 1<sup>st</sup> Order Correction Option**

$Temp_{Adj}(\Delta VDDA) = Corr_{Voltage} = b_1(\Delta VDDA)$

**V Corr Method = 2, TempCode = Corr<sub>Voltage</sub> 2<sup>nd</sup> Order Correction Option**

The B Mean term is the nominal  $VDDA$  Code value when 3.3V is applied to  $VDDA$ :

$$VDDACode_{Nom} = VDDACode \text{ measured by IC when } 3.3 \text{ V is applied to } VDDA$$

In this example, that value is 2330.

Because the supply voltage range of the TMx00 is specified at  $\pm 5\%$ , the standard deviation of the  $VDDA$  Code range is set internally with respect to the nominal  $VDDA$  Code.

$$VDDA_{SD} = \text{Standard deviation of } VDDA$$

The Standard Deviation is defined so a 5% high supply voltage gives a  $VDDA_{SD}$  (or  $\sigma$ ) of +3.0, and a 5% low supply gives a  $VDDA_{SD}$  of -3.0.

$$VDDA_{SD} = VDDACode_{NOM} \left( \frac{0.05}{3} \right)$$

$VDDACode_{NOM}$  is the ADC code value measured by the IC when the VDDA supply voltage is 3.3V. A typical ADC code value is 2330. Note that the VDDA ADC input is divided by 2, so the 2330 code represents an input value of 1.65V.

$$\Delta VDDA = \frac{VDDACode_{Meas} - VDDACode_{Nom}}{VDDA_{SD}}$$

$VDDA_{SD}$  = Standard deviation of VDDA set so that  $\pm 0.05$  wrt  $VDDACode_{3.3V}$  produces a  $\sigma$  of  $\pm 3$

$$VDDA_{SD} = VDDACode_{NOM} \left( \frac{0.05}{3} \right)$$

$$\Delta_{VDDA} = \frac{VDDACode_{Meas} - VDDACode_{Nom}}{VDDA_{SD}}$$

If the frequency variation is mostly caused by the variation in temperature sensor reading, the B1 term input should be set to the inverse of the *Temp Code* slope:

$$B1 = -\frac{1}{m}$$

$$m = \frac{(TempCode_2 - TempCode_1)}{(VDDACode_2 - VDDACode_1)}$$

For this example:

$$m = \frac{(2550 - 2542)}{(2446 - 2213)} = 0.034335$$

$$B1 = -\frac{1}{m} = -\frac{1}{0.034335} = -29.125$$

If the frequency variation consists of terms dependent on the temperature sensor and the VDDA supply voltage, then the B1 should be chosen so that the frequency slope is minimized over the supply voltage range. For example, this method of adjustment would be needed for applications in which BYPASS and VDDA are connected. The DAC output voltage changes with the BYPASS value so the needed tuning command voltage would also change.

**HEXIUS SEMICONDUCTOR**

General Setup Oscillator Setup Correction Setup THRM Calc Thermal Setup User Block

### Frequency Correction

**Correction Technique ?**

- ☐ Disabled
- ☐ Disabled with V/T Monitoring
- ☒ Polynomial
- ☐ Lookup Table

Correction DAC: 1000 0.000 V

**Correction Update Intervals ?**

Temperature Sample Interval: 20 ms  
 Frequency Correction Interval: 30 ms (30 ms)  
 Averaging Intervals (Freq Corr): 10

**Temp Code Source ?**

- ☒ From Internal Temp Sensor
- ☐ From THRM

**Measure ?**

Measure Temp, Volts & Correction		
Measured IC Temp	26.9	C
IC Temp Code	2548	
THRM	2827	
<hr/>		
Correction DAC Code	1000	
Correction DAC Volts	0	V
<hr/>		
VDDA Code	2338	
VDDA Voltage	3.311	V

### Polynomial Correction

**Coefficients ?**

Correction Order: 7

A0	1.3933
A1	0.3158
A2	0.0034
A3	-0.2854
A4	0.0446
A5	0.0517
A6	-0.0179
A7	-0.0128
A8	
A9	
Mean	2544.9333
Std Dev	327.2820

**Voltage Corr ?**

V Corr Method: 1

B1	-29.125
B2	0
B3	0
C1	0
C2	0
C3	0
B Mean	2330

### Lookup Table Correction

**Lookup Table ?**

	Temp Code	DAC Corr
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0

**Information**

### Voltage Correction

V Corr Method defines an optional enhanced correction algorithm that generates an adjustment term, compensating for variations over supply voltage (VDDA) or supply voltage over temperature. Several different correction methods are available, defined by the value of 'V Corr Order'.

$\Delta VDDA = VDDA \text{ normalized to Mean and Standard Deviation}$ .  
 Mean (B\_Mean) is the VDDA Code for 3.3V applied to the VDDA pin. The Standard Deviation is computed internally to equal  $\pm 3$  (3σ) when the supply is 3.3V  $\pm 5\%$ .

$temp\_code = \text{The value from the Temp Code Source (Internal IC Sensor or THRM)}$ .

### TempCode Adjustment Option

$\Delta VDDA = VDDA \text{ normalized to Mean and Standard Deviation}$ .  
 Mean (B\_Mean) is the VDDA Code for 3.3V applied to the VDDA pin. The Standard Deviation is computed internally to equal  $\pm 3$  (3σ) when the supply is 3.3V  $\pm 5\%$ .

$temp\_code = \text{The value from the Temp Code Source (Internal IC Sensor or THRM)}$ .

$\Delta TempCode = TempCodeNormalized + Temp_{ADJ}(\Delta VDDA)$

$\Delta TempCode$  is the new  $TempCodeNormalized$  value input into the main  $Corr_{TEMP}(X)$  function for frequency correction.

**V Corr Method = 0**

No correction applied.

**V Corr Method = 1, TempCode=Corr\_Voltage 1<sup>st</sup> Order Correction Option**

$Temp_{ADJ}(\Delta VDDA) = Corr_{Voltage} = b_1(\Delta VDDA)$

**V Corr Method = 2, TempCode=Corr\_Voltage 2<sup>nd</sup> Order Correction Option**

---

IC Part Number: TM200  
 Firmware Rev: 1.8

**Connect, Restart Info ?**

Connect to IC [ ]  
 Restart IC [ ]

(62) Receiving data from device  
 RX Data: 18 09 DF  
 (63) Transmitting data to device  
 TX Data: 1C E4  
 (63) Receiving data from device  
 RX Data: 1C 03 E8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F9  
 (64) Transmitting data to device  
 TX Data: 14 05 40 A7  
 (64) Receiving data from device  
 RX Data: 14 00 EC  
 (65) Transmitting data to device  
 TX Data: 14 00 EC

**Load or Save IC Setup ?**

[ Load IC Setup from File ]  
 [ Save IC Setup to File ]

**Send Setup to IC (RAM based) ?**

[ Send Current Setup Tab Items to IC ]  
 [ Send All Setup Tab Items to IC ]

**Load IC OTP Setup ?**

[ Load Stored IC OTP Setup ]

**Store Setup into IC OTP ?**

[ Store Current Setup into IC OTP ]

Remaining Setup Stores: 5

Comms Established

The TMx00 will calculate all the necessary normalization values to generate and apply the *Temp Code* adjustment term

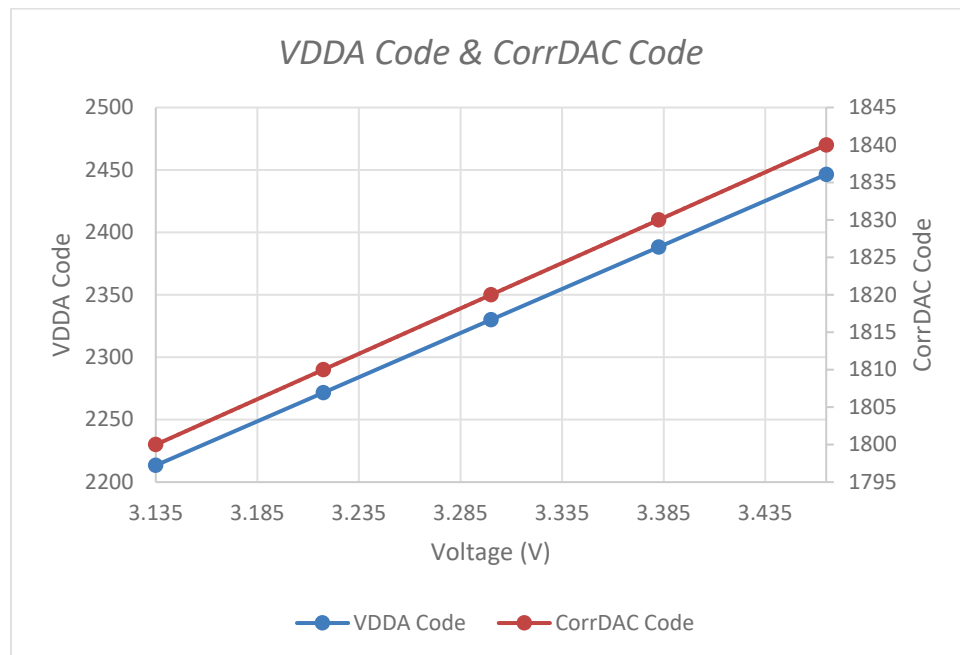
$$\Delta TempCode(\Delta VDDA) = Corr_{voltage}(\Delta VDDA) = b_1(\Delta VDDA)$$

### CorrDAC Code Adjustment Example - Linear Correction

The *CorrDAC Code* adjustment is identical to the *Temp Code* adjustment but uses *CorrDAC Codes* instead of *Temp Codes*. If the frequency output shift of the module is not solely dependent on *Temp Code* shifts, the collected data be changed to the *VDDA Code* and *CorrDAC Code* for a correction implementation. This implementation may be a good technique if the *BYPASS* voltage is tied to *VDDA* which will cause a shift in the internal Varicap (if used). The data below is from a supply voltage sweep that collects the *VDDA Code* and *CorrDAC Code* information needed for a correction implementation.

**Table 8 Voltage Correction CorrDAC Code Data Set Example**

<i>Supply (V)</i>	<i>VDDA Code</i>	<i>CorrDAC Code</i>
3.135	2213	1816
3.2175	2272	1818
3.30	2330	1820
3.3825	2388	1822
3.465	2446	1824



The relationship between *VDDA Code* and *CorrDAC Code* is linear. Therefore, select V Corr Method 5 which is:

CorrDAC Code Corr<sub>voltage</sub> 1<sup>st</sup> Order Correction Option

$$\Delta \text{CorrDAC Code}(\Delta VDDA) = \text{Corr}_{\text{voltage}}(\Delta VDDA) = b_1(\Delta VDDA)$$

B Mean is 2330

The B1 term input needs to be inverse of the *CorrDAC Code* slope:

$$B1 = -\frac{1}{m}$$

$$m = \frac{(\text{CorrDAC Code}_2 - \text{CorrDAC Code}_1)}{(VDDACode_2 - VDDACode_1)}$$

For this example:

$$m = \frac{(1840 - 1800)}{(2446 - 2213)} = 0.171674$$

$$B1 = -\frac{1}{m} = -\frac{1}{0.171674} = -5.825$$

**Hexius Semiconductor TMx00 Control Program**

Options Help

**HEXIUS SEMICONDUCTOR**

General Setup Oscillator Setup **Correction Setup** THRM Calc Thermal Setup User Block

**Frequency Correction**

Correction Technique

- ☐ Disabled
- ☐ Disabled with V/T Monitoring
- ☒ Polynomial
- ☐ Lookup Table

Correction DAC: 1200 0.000 V

**Correction Update Intervals**

Temperature Sample Interval: 20 ms

Frequency Correction Interval: 30 ms (30 ms)

Averaging Intervals: 10 (Freq Corr)

**Temp Code Source**

- ☒ From Internal Temp Sensor
- ☐ From THRM

**Measure**

Measure Temp, Volts & Correction	
Measured IC Temp	25.9 C
IC Temp Code	2540
THRM	2757
Correction DAC Code	1200
Correction DAC Volts	0 V
VDDA Code	2213
VDDA Voltage	3.134 V

**Polynomial Correction**

Coefficients

Correction Order: 7

A0: 1.3933

A1: 0.3158

A2: 0.0034

A3: -0.2854

A4: 0.0446

A5: 0.0517

A6: -0.0179

A7: -0.0128

A8:

A9:

Mean: 2544.9333

Std Dev: 327.2820

**Voltage Corr**

V Corr Method: 5

B1: -5.825

B2: 0

B3: 0

C1: 0

C2: 0

C3: 0

B Mean: 2330

**Lookup Table Correction**

Lookup Table

Temp Code	DAC Corr
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0

**Information**

$b_1(\Delta VDDA) = c_1(\Delta VDDA)(\text{temp\_code}) + c_2(\Delta VDDA)^2(\text{temp\_code}) + c_3(\Delta VDDA)^3(\text{temp\_code})^2$

**CorrDAC Code Adjustment Option**

Applying the correction by adjusting the *CorrDACCode* is defined as:

$\Delta \text{CorrDACCode} = \text{CorrDACCode} + \text{CorrDAC}_{\text{adj}}(\Delta VDDA)$   
(lookup table correction)

Or

$\Delta \text{CorrDACVoltage} = \text{CorrDACVoltage} + \text{CorrDAC}_{\text{adj}}(\Delta VDDA)$   
(polynomial correction)

Where:

$\Delta \text{CorrDACCode}$  is the new *CorrDACCode* value output for frequency correction (lookup table correction).

$\Delta \text{CorrDACVoltage}$  is the new *CorrDACVoltage* value output for frequency correction (polynomial correction).

*CorrDAC<sub>adj</sub>* is the *CorrDACCode* or *CorrDACVoltage* adjustment based on voltage and temperature measurements.

**V Corr Method = 4**

No correction applied.

**V Corr Method = 5, CorrDACCode=Corr<sub>voltage</sub> 1<sup>st</sup> Order Correction Option**

$\text{CorrDAC}_{\text{adj}}(\Delta VDDA) = \text{Corr}_{\text{voltage}}(\Delta VDDA) = b_1(\Delta VDDA)$

**V Corr Method = 6, CorrDACCode=Corr<sub>voltage</sub> 2<sup>nd</sup> Order Correction Option**

$\text{CorrDAC}_{\text{adj}}(\Delta VDDA) = \text{Corr}_{\text{voltage}}(\Delta VDDA) = b_1(\Delta VDDA) + b_2(\Delta VDDA)^2 + b_3(\Delta VDDA)^3$

**V Corr Method = 7, CorrDACCode=Corr<sub>voltage</sub> & 3<sup>rd</sup> Order Correction Option**

IC Part Number: TM200

Firmware Rev: 1.8

**Connect, Restart Info**

Connect to IC

Restart IC

(211) Transmitting data to device  
TX Data: 14 08 40 A4  
(211) Receiving data from device  
RX Data: 14 00 EC  
(212) Transmitting data to device  
TX Data: 15 EB  
(212) Receiving data from device  
RX Data: 15 A5 08 08 36  
(213) Transmitting data to device  
TX Data: 1F E1  
(213) Receiving data from device  
RX Data: 1F 3D F3 B6 46 C3 8A 2B 45 09

Comms Established

**Load or Save IC Setup**

Load IC Setup from File

Save IC Setup to File

**Send Setup to IC (RAM based)**

Send Current Setup Tab Items to IC

Send All Setup Tab Items to IC

**Load IC OTP Setup**

Load Stored IC OTP Setup

**Store Setup into IC OTP**

Store Current Setup into IC OTP

Remaining Setup Stores: 5

## 8 REFERENCES

*TM100 Datasheet*

*TM200 Datasheet*

*TMx00 Control Software & EVB Kit Guide*

*TMx00 Programming Reference Manual*