Application Note

CMX649 Recommended Settings

AN/2WR/649_Settings/2 February 2013

Additional Resources	None

Introduction

The CMX649 voice codec offers the designer multiple voice coding options for a wide range of applications. While its flexibility lends itself to many different applications, the large number of possible CMX649 operating configurations can cause confusion. The purpose of this document is to alleviate this confusion by presenting optimised CMX649 register settings for various sampling rates.

The CMX649 data sheet should be consulted during the review of this document.

History

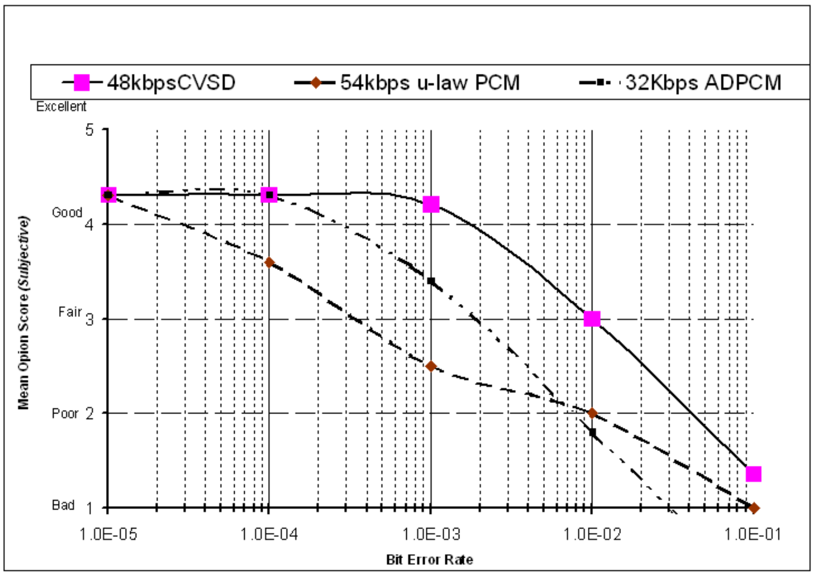
Version	Changes	Date
2.0	XTAL/CLOCK low frequency input limits included	20-02-13
1.0	Original Release	3-11-03

TABLE OF CONTENTS

1 Overview

While the CMX649 can successfully serve in a wide range of voice coding applications, this innovative device is optimally positioned for wireless applications. Adaptive delta modulation (ADM), of which continuously variable slope delta modulation (CVSD) is a subset, is an ideal voice coding scheme for wireless applications due to its robust performance in the presence of bit errors.

The superiority of ADM/CVSD for wireless applications is demonstrated in the following figure:



The x-axis of this graph represents bit error rate; movement to the right on the x-axis represents an increasing number of bit errors, and consequently, a poorer quality signal. The y-axis represents "mean opinion score" (MOS), a subjective assessment of the recovered audio quality after encode/decode processing. Larger MOS scores translate to better audio quality as perceived by the listener. (A "MOS" score of three is considered the minimum for "toll quality" speech.) As can be seen from this graph, CVSD (and thus ADM) maintains better voice quality with a lower data rate than does PCM or ADPCM in a poor BER channel.

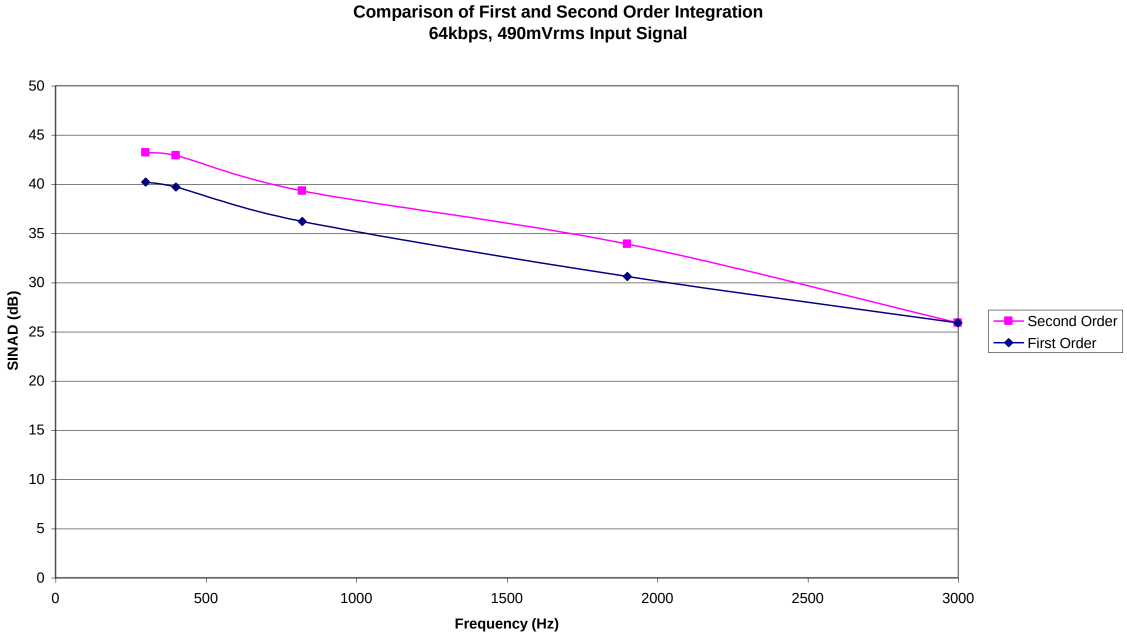
Since ADM is an outstanding voice coding technique for wireless applications, and since the CMX649 is ideally positioned to serve wireless applications, the register settings provided in this document are developed for the use of ADM voice coding.

2 First Order Integration vs Second Order Integration

"First order" integration refers to the use of a single integrator in the encoder feedback path. This technique is used with CVSD coding and it provides good voice quality. The addition of a second integration stage in the feedback path, referred to as "second order" integration, provides an improvement in voice quality over first order integration.

The difference between first order integration and second order integration comes down to fidelity at higher amplitude signals. Second order integration yields better fidelity for higher amplitude and higher frequency signals, but it does so at the expense of amplitude response on higher frequency signals. First order integration provides a flatter frequency response, lower idle channel noise, and somewhat higher distortion than does second order integration.

The following plot illustrates the difference in SINAD performance, for a given sampling rate, for first and second order integration.



3 Considerations for Optimal Voice Quality

3.1 Companding Rule Selection

Normally, the microphone input signal is compared to its predicted value in the CMX649 encoder. The encoder comparator outputs a logic one if the microphone input exceeds the predicted value, while a logic zero is output otherwise. The comparator output signal is then fed to the encoder delay register.

Changes of input signal amplitude that do not exceed the current quantization step size cause the comparator to output alternating ones and zeros. Rapidly changing input signals, however, can exceed the quantization step size and cause the encoder to be in a “slew-rate limited” condition (aka “slope overload”). When this happens, the ADM representation cannot change fast enough to keep up with the microphone input signal, and the comparator produces a string of ones or zeros as a result of the slope overload condition.

The encoder delay register allows from three to six bit times of slope overload to occur before the step size is changed. This time period, also called the “companding rule”, is programmed with bits 9-8 of the DECODE ($D1) and ENCODE ($E1) ADM CONTROL registers.

For example, with the companding rule set to “3 of 3”, a string of three consecutive ones or zeros from the comparator will be detected as slope overload. Once the slope overload condition has been detected, the encoder will adjust the quantization step size so that the ADM representation can more closely track the analogue input signal.

3.2 Quantization Step Height Selection

The quantization step heights are programmed with bits 12-10 of the DECODE ($D1) and ENCODE ($E1) ADM CONTROL registers.

Two problems can occur if the maximum and minimum step heights are improperly selected; slope overload and granular noise.

Slope overload occurs when the step size is too small; the digital representation cannot adequately track changes in the analogue input. Granular noise occurs when the step size is too large; input signal changes smaller than the step size are not digitised. Granular noise can occur for any step size, so it is advantageous to keep the step size as low as possible while providing adequate protection from slope overload.

Two parameters of concern with any voice coding scheme are the decoded signal level and SINAD. The following considerations describe the impact of quantization step height selection on both of these important parameters.

SINAD Considerations:

1. In general, SINAD drops as input frequency increases. This degradation is due to slope overload, which is maximised with higher input signal amplitude and frequency. Slope overload degrades the SINAD measurement.
2. Offset compensation improves SINAD at mid & upper frequencies, but produces a SINAD hit at low frequencies.
3. For a given step height with higher input frequencies (e.g. 2kHz), SINAD is higher for lower input amplitude than for higher input amplitude.
4. For a large input amplitude at high input frequencies, large max and min step height yields the best SINAD. This is because the larger step heights help counteract slope overload better than small step heights.
5. For a large input amplitude at low input frequencies, large min step yields the best SINAD. Slope overload isn't as much of a problem with low frequencies because the codec has time to keep up with the signal. With large input signals, there will be fewer instances of threshold effects, so granular noise is reduced. The max step height selection is not critical to SINAD performance in this situation.
6. For a small input amplitude, **small** min step yields best SINAD across all input frequencies. This is because a small step height can easily keep up with input changes without adding too much granular noise. The size of the max step height for small input amplitudes is not critical to SINAD performance.

Decoded Signal Level Considerations:

1. For small input signals, a small minimum step height will maximise the decoder output level for all input frequencies.
2. For large input signals, a large max/min step height will maximise the decoder output level across all input frequencies.

These SINAD and decoded signal level recommendations can be summarized in the following table. These recommendations are general in nature; optimal step height settings are application dependent:

Max and Min Step Heights for Optimal SINAD and Decoded Signal Level Performance	**Input Level at Encoder**			
	Low (e.g. 50mVrms)		High (e.g. 490mVrms)	
	Min	Small	Min	Large
	Max	Not critical	Max	Large
	Min	Small	Min	Large
	Max	Not critical	Max	Large

Table 1: Summary of Recommendations for Max & Min Step Heights

3.3 Integrator Time Constant Selection

The estimator integrator is used in the encoder/decoder feedback path. The syllabic integrator adjusts the quantization step height. The time constant of the syllabic integrator determines how quickly the step height can increase or decrease. The shorter the time constant, the faster the step height can be changed.

The syllabic integrator typically has a time constant that is much longer (e.g. 20x to 30x) than the estimator integrator. For example, for Bluetooth compatibility at 64kbps:

- Syllabic integrator time constant = 16ms
- Estimator integrator time constant = 0.5ms

Typical values for syllabic integrator time constants are in the range of 5-10ms, but optimal values are application dependent.

The time constants for the syllabic and estimator integrators are selected in the DECODE ($D1) and ENCODE ($E1) ADM CONTROL register.

3.4 Second Order Integration

ADM uses an estimator integrator in the encoder feedback path. While this approach yields good voice quality, additional improvement in voice quality can be achieved by adding an additional integration stage to the estimator integrator. This process, known as “second order integration”, introduces a second integrator in the estimator integration to create a smoother reconstructed signal at the decoder.

Second order integration is enabled or disabled with bits 4-3 of the DECODE ($D1) and ENCODE ($E1) ADM CONTROL registers.

When second order integration is used, the encoder can experience instability and oscillations. A “zero” can be added to the estimator integrator transfer function to improve stability with second order integration. The settings for zero selection are contained within bits 2-1 of the ENCODE ($E1) ADM CONTROL register. For example, with a 64kbps data rate:

Encode ADM Control Register ($E1)		**Zero Frequency for 64kbps Data Rate**
Bit 2	Bit 1	
0	0	N/A (first order estimator)
0	1	42.7kHz
1	0	25.6kHz
1	1	14.2kHz

Table 2: Example of Zero Frequency Determination

4 CMX649 Register Settings

4.1 General Comments

The objective of “good voice quality” is highly subjective; what is “good” for one listener may be poor for another. With this in mind, settings for both first and second order integration are provided so that the reader can experiment and obtain an optimal response for his/her application.

These settings have been tested in a laboratory environment and found to produce satisfactory voice quality. It is expected that these settings will result in satisfactory voice quality in realistic operating conditions as well, and the reader is encouraged to experiment with these settings and explore possible enhancements for specific applications.

Settings are provided for sampling rates of 16kbps, 24kbps, 32kbps, 64kbps, and 128kbps. Higher data rates provide higher fidelity speech, at the expense of greater transmitted bandwidth.

The format for the settings is:

```
*****************
Register Address {hex}  Register Contents {hex}
        //Discussion of items selected by register contents
*****************
```

For example, this portion of pseudocode:

```
*****************
61      00
        //Setup analogue section - filters set for 2.9KHz mode
*****************
```

…results in register 61h (AAF/AIF Bandwidth) receiving a value of 00h, which then configures the AAF and AIF filters in the CMX649 analogue section for 2.9kHz mode.

Sections of pseudocode with only a “01” represent a “general reset” command, a single byte C-BUS transaction that resets the CMX649. This command is executed by transmitting a “01” to the CMX649 over its “Command Data” line; no register address is necessary with this command. Please review the CMX649 data sheet, if necessary, for more information about this command.

4.2 First Order Integration Settings

4.2.1 16kbps

```
01
        // initialise device with general reset
        // this powers down everything excluding the xtal oscillator circuit

61      00
        // setup analogue section - filters set for 2.9KHz mode

62      BE
        // volume=0dB, sidetone= -21dB and off

63      80
        // audio level=0dB

64      55
65      55
        // power control, everything on low

70      00
        // codec mode=ADM unbuffered (continuous bit serial mode)

71      0000
        // data scrambler and descrambler both off

72      E9C0
        // using 4.096MHz master clock (NOTE: If a Xtal is to be used it should be a >8MHz
        device.)
        // filter clock prescaler divider=2, filter clock main divider=8 => 256kHz internal SCF clock
   // bit clock prescaler divider=4, encode and decode bit clock dividers=1, with constant divider=64
   => 16kbps sampling rate

73      0078
        // PLL is off, bypass PLL data filter and power it down
        // Internal decode and encode clocks from decoder internal clock (synchronised to XTAL/CLK
        input)

81      88
        // enable encoder and decoder with no IRQs

D0      00B8
        // decimate by 8
        // decode ADM input from RX Data
        // ADM estimator drives output
        // VAD attack tc=4ms and decay tc=128ms
        // normal VAD outputs

D1      0940
E1      0940
        // ADM mode syllabic tc=10.7ms
        // step size dynamic range 5120/20
        // companding rule = 4 of 4
```

// principal fc=239Hz
// second order fc=N/A
// encoder zero fc=N/A decoder zero fc=N/A
// decoder zero at 8kHz i.e. bit_rate/2 disabled

D2 0200
E2 0200
// VAD thresholds ~20mv

D8 AA
E8 AA
// prime idle pattern into C-BUS ADM input registers

E0 01B8
// ADM encode feedback from comparator, idle channel enhancement active, otherwise as decoder

E3 0020
// load a small positive constant into encoder offset input reg to enable idle channel enhancement

4.2.2 24kbps

01
// initialise device with general reset
// this powers down everything excluding the xtal oscillator circuit

61 00
// setup analogue section - filters set for 2.9KHz mode

62 BE
// volume=0dB, sidetone= -21dB and off

63 80
// audio level=0dB

64 55
65 55
// power control, everything on low

70 00
// codec mode=ADM unbuffered (continuous bit serial mode)

71 0000
// data scrambler and descrambler both off

72 E91B
// using 4.096MHz master clock **(NOTE: If a Xtal is to be used it should be a >8MHz device.)**
// filter clock prescaler divider =2, filter clock main divider=8 => 256kHz internal SCF clock

// bit clock prescaler divider=1, encode and decode bit clock dividers=2.625, since constant divider=64 => 24kbps sampling rate

73 0078
// PLL is off, bypass PLL data filter and power it down
// internal decode and encode clocks from decoder internal clock (synchronised to XTAL/CLK input)

81 88
// enable encoder and decoder with no IRQs

D0 00B8
// decimate by 8
// decode ADM input from RX Data
// ADM estimator drives output
// VAD attack tc=4ms and decay tc=128ms
// normal VAD outputs

D1 2960
E1 2960
// ADM mode syllabic tc=10.7ms
// step size dynamic range 5120/20
// companding rule = 4 of 4
// principal fc=239Hz
// second order fc=N/A
// encoder zero fc=N/A decoder zero fc=N/A
// decoder zero at 12kHz i.e. bit_rate/2 disabled

D2 0200
E2 0200
// VAD thresholds ~20mv

D8 AA
E8 AA
// prime idle pattern into C-BUS ADM input registers

E0 01B8
// ADM encode feedback from comparator, idle channel enhancement active, otherwise as decoder

E3 0020
// load a small positive constant into encoder offset input reg to enable idle channel enhancement

4.2.3 32kbps

01
// initialise device with general reset
// this powers down everything excluding the xtal oscillator circuit

61 11
// setup analogue section - filters set for 3.7KHz mode

62 BE
// volume=0dB, sidetone= -21dB and off

63 80
// audio level=0dB

64 55
65 55
// power control, everything on low

70 00
// codec mode=ADM unbuffered (continuous bit serial mode)

71 0000
// data scrambler and descrambler both off

72 E940

// using 4.096MHz master clock **(NOTE: If a Xtal is to be used it should be a >8MHz device.)**
// filter clock prescaler divider=2, filter clock main divider=8 => 256kHz internal SCF clock
// bit clock prescaler divider=2, encode and decode bit clock dividers=1, since constant divider=64 => 32kbps sampling rate

73 0078
// PLL is off, bypass PLL data filter and power it down
// internal decode and encode clocks from decoder internal clock (synchronised to XTAL/CLK input)

81 88
// enable encoder and decoder with no IRQs

D0 00B8
// decimate by 8
// decode ADM input from RX Data
// ADM estimator drives output
// VAD attack tc=4ms and decay tc=128ms
// normal VAD outputs

D1 4980
E1 4980
// ADM mode syllabic tc=10.7ms
// step size dynamic range 5120/20
// companding rule = 4 of 4
// principal fc=239Hz
// second order fc=N/A
// encoder zero fc=N/A decoder zero fc=N/A
// decoder zero at 16kHz i.e. bit_rate/2 disabled

D2 0200
E2 0200
// VAD thresholds ~20mv

D8 AA
E8 AA
// prime idle pattern into C-BUS ADM input registers

E0 01B8
// ADM encode feedback from comparator, idle channel enhancement active, otherwise as decoder

E3 0020
// load a small positive constant into encoder offset input reg to enable idle channel enhancement

4.2.4 64kbps

01
// initialise device with general reset
// this powers down everything excluding the xtal oscillator circuit

61 11
// setup analogue section - filters set for 3.7KHz mode

62 BE
// volume=0dB, sidetone= -21dB and off

63 80
// audio level=0dB

64 55
65 55
// power control, everything on low

70 00
// codec mode=ADM unbuffered (continuous bit serial mode)

71 0000
// data scrambler and descrambler both off

72 E900
// using 4.096MHz master clock **(NOTE: If a Xtal is to be used it should be a >8MHz device.)**
// filter clock prescaler divider=2, filter clock main divider=8 => 256kHz internal SCF clock
// bit clock prescaler divider=1, encode and decode bit clock dividers=1, since constant divider=64 => 64kbps sampling rate

73 0078
// PLL is off, bypass PLL data filter and power it down
// internal decode and encode clocks from decoder internal clock (synchronised to XTAL/CLK input)

81 88
// enable encoder and decoder with no IRQs

D0 00B8
// decimate by 8
// decode ADM input from RX Data
// ADM estimator drives output
// VAD attack tc=4ms and decay tc=128ms
// normal VAD outputs

D1 89C0
E1 89C0
// ADM mode syllabic tc=10.7ms
// step size dynamic range 5120/20
// companding rule = 4 of 4
// principal fc=239Hz
// second order fc=N/A
// encoder zero fc=N/A decoder zero fc=N/A
// decoder zero at 32kHz i.e. bit_rate/2 disabled

D2 0200
E2 0200
// VAD thresholds ~20mv

D8 AA
E8 AA
// prime idle pattern into C-BUS ADM input registers

E0 00B8
// ADM encode feedback from comparator, otherwise as decoder

E3 0000
// clear encoder offset input reg to disable idle channel enhancement

4.2.5 128kbps

01
// initialise device with general reset
// this powers down everything excluding the xtal oscillator circuit

61 33
// setup analogue section - filters set for 7.0KHz mode

62 BE
// volume=0dB, sidetone= -21dB and off

63 80
// audio level=0dB

64 55
65 55
// power control, everything on low

70 00
// codec mode=ADM unbuffered (continuous bit serial mode)

71 0000
// data scrambler and descrambler both off

72 EC00
// using 8.0MHz master clock **(NOTE: this is a different master clock frequency than those used for previous settings.)**
// filter clock prescaler divider=4, filter clock main divider=8.0 => 256kHz internal SCF clock

// bit clock prescaler divider=1, encode and decode bit clock dividers=1, since constant divider=64 => 128kbps sampling rate

73 0078
// PLL is off, bypass PLL data filter and power it down
// internal decode and encode clocks from decoder internal clock (synchronised to XTAL/CLK input)

81 88
// enable encoder and decoder with no IRQs

D0 00B8
// setup decoder
// decimate by 8
// decode ADM input from RX Data
// ADM estimator drives output
// VAD attack tc=4ms and decay tc=128ms
// normal VAD outputs

D1 C9E0
E1 C9E0
// ADM mode syllabic tc=10.7ms
// step size dynamic range 5120/20
// companding rule = 4 of 4
// principal fc=318Hz
// second order fc=none
// encoder zero fc=N/A decoder zero fc=N/A
// decoder zero at 64kHz i.e. bit_rate/2 disabled

D2 0200

E2 0200
// VAD thresholds ~20mv

D8 AA
E8 AA
// prime idle pattern into C-BUS ADM input registers

E0 00B8
// ADM encode feedback from comparator, otherwise as decoder

E3 0000
// clear encoder offset input reg to disable idle channel enhancement

4.3 Second Order Integration Settings

4.3.1 16kbps

01
// initialise device with general reset
// this powers down everything excluding the xtal oscillator circuit

61 00
// setup analogue section - filters set for 2.9KHz mode

62 BE
// volume=0dB, sidetone=-21dB and off

63 80
// audio level=0dB

64 55
65 55
// power control, everything on low

70 00
// codec mode=ADM unbuffered (continuous bit serial mode)

71 0000
// data scrambler and descrambler both off

72 E9C0
// using 4.096MHz master clock **(NOTE: If a Xtal is to be used it should be a >8MHz device.)**
// filter clock prescaler divider=2, filter clock main divider=8 => 256kHz internal SCF clock
// bit clock prescaler divider=4, encode and decode bit clock dividers=1, with constant divider=64 => 16kbps sampling rate

73 0078
// PLL is off, bypass PLL data filter and power it down
// internal decode and encode clocks from decoder internal clock (synchronised to XTAL/CLK input)

81 88
// enable encoder and decoder with no IRQs

D0 00B8
// setup decoder
// decimate by 8

// decode ADM input from RX Data
// ADM estimator drives output
// VAD attack tc=4ms and decay tc=128ms
// normal VAD outputs

D1 0951
E1 0952
// ADM mode syllabic tc=10.7ms
// step size dynamic range 5120/20
// companding rule = 4 of 4
// principal fc=239Hz
// second order fc=955Hz
// encoder zero fc=1.7KHz decoder zero fc=N/A
// decoder zero at 8kHz i.e. bit_rate/2 enabled

D2 0200
E2 0200
// VAD thresholds ~20mv

D8 AA
E8 AA
// prime idle pattern into C-BUS ADM input registers

E0 01B8
// ADM encode feedback from comparator, idle channel enhancement active, otherwise as decoder

E3 0020
// load a small positive constant into encoder offset input reg to enable idle channel enhancement

4.3.2 24kbps

01
// initialise device with general reset
// this powers down everything excluding the xtal oscillator circuit

61 00
// setup analogue section - filters set for 2.9KHz mode

62 BE
// volume=0dB sidetone=-21dB and off

63 80
// audio level=0dB

64 55
65 55
// power control everything on low

70 00
// codec mode = ADM unbuffered (continuous bit serial mode)

71 0000
// data scrambler and descrambler both off

72 E91B
// using 4.096MHz master clock **(NOTE: If a Xtal is to be used it should be a >8MHz device.)**
// filter clock prescaler divider =2, filter clock main divider=8 => 256kHz internal SCF clock

// bit clock prescaler divider=1, encode and decode bit clock dividers=2.625, since constant divider=64 => 24kbps sampling rate

73 0078
// PLL is off, bypass PLL data filter and power it down
// internal decode and encode clocks from decoder internal clock (synchronised to XTAL/CLK input)

81 88
// enable encoder and decoder with no IRQs

D0 00B8
// setup decoder
// decimate by 8
// decode ADM input from RX Data
// ADM estimator drives output
// VAD attack tc=4ms and decay tc=128ms
// normal VAD outputs

D1 2979
E1 297A
// ADM mode syllabic tc=10.7ms
// step size dynamic range 5120/20
// companding rule = 4 of 4
// principal fc=239Hz
// second order fc=1910Hz
// encoder zero fc=3.4KHz decoder zero fc=N/A
// decoder zero at 12kHz i.e. bit_rate/2 enabled

D2 0200
E2 0200
// VAD thresholds ~20mv

D8 AA
E8 AA
// prime idle pattern into C-BUS ADM input registers

E0 01B8
// ADM encode feedback from comparator, idle channel enhancement active, otherwise as decoder

E3 0020
// load a small positive constant into encoder offset input reg to enable idle channel enhancement

4.3.3 32kbps

01
// initialise device with general reset
// this powers down everything excluding the xtal oscillator circuit

61 11
// setup analogue section - filters set for 3.7KHz mode

62 BE
// volume=0dB sidetone=-21dB and off

63 80
// audio level=0dB

64 55
65 55
// power control, everything on low

70 00
// codec mode=ADM unbuffered (continuous bit serial mode)

71 0000
// data scrambler and descrambler both off

72 E940
// using 4.096MHz master clock **(NOTE: If a Xtal is to be used it should be a >8MHz device.)**
// filter clock prescaler divider=2, filter clock main divider=8 => 256kHz internal SCF clock
// bit clock prescaler divider=2, encode and decode bit clock dividers=1, since constant divider=64 => 32kbps sampling rate

73 0078
// PLL is off, bypass PLL data filter and power it down
// internal decode and encode clocks from decoder internal clock (synchronised to XTAL/CLK input)

81 88
// enable encoder and decoder with no IRQs

D0 00B8
// setup decoder
// decimate by 8
// decode ADM input from RX Data
// ADM estimator drives output
// VAD attack tc=4ms and decay tc=128ms
// normal VAD outputs

D1 4999
E1 499A
// ADM mode syllabic tc=10.7ms
// step size dynamic range 5120/20
// companding rule = 4 of 4
// principal fc=239Hz
// second order fc=1910Hz
// encoder zero fc=3.4KHz decoder zero fc=N/A
// decoder zero at 16kHz i.e. bit_rate/2 enabled

D2 0200
E2 0200
// VAD thresholds ~20mv

D8 AA
E8 AA
// prime idle pattern into C-BUS ADM input registers

E0 01B8
// ADM encode feedback from comparator, idle channel enhancement active, otherwise as decoder

E3 0020
// load a small positive constant into encoder offset input reg to enable idle channel enhancement

4.3.4 64kbps

01
// initialise device with general reset
// this powers down everything excluding the xtal oscillator circuit

61 11
// setup analogue section - filters set for 3.7KHz mode

62 BE
// volume=0dB, sidetone= -21dB and off

63 80
// audio level=0dB

64 55
65 55
// power control everything on low

70 00
// codec mode=ADM unbuffered (continuous bit serial mode)

71 0000
// data scrambler and descrambler both off

72 E900
// using 4.096MHz master clock **(NOTE: If a Xtal is to be used it should be a >8MHz device.)**
// filter clock prescaler divider=2, filter clock main divider=8 => 256kHz internal SCF clock
// bit clock prescaler divider=1, encode and decode bit clock dividers=1, since constant divider=64 => 64kbps sampling rate

73 0078
// PLL is off, bypass PLL data filter and power it down
// internal decode and encode clocks from decoder internal clock (synchronised to XTAL/CLK input)

81 88
// enable encoder and decoder with no IRQs

D0 00B8
// setup decoder
// decimate by 8
// decode ADM input from RX Data
// ADM estimator drives output
// VAD attack tc=4ms and decay tc=128ms
// normal VAD outputs

D1 89B9
E1 89BC
// ADM mode syllabic tc=10.7ms
// step size dynamic range 5120/20
// companding rule = 4 of 4
// principal fc=239Hz
// second order fc=1910Hz
// encoder zero fc=3.4KHz decoder zero fc=N/A
// decoder zero at 32kHz i.e. bit_rate/2 enabled

D2 0200
E2 0200
// VAD thresholds ~20mv

D8 AA
E8 AA
// prime idle pattern into C-BUS ADM input registers

E0 00B8
// ADM encode feedback from comparator, otherwise as decoder

E3 0000
// clear encoder offset input reg to disable idle channel enhancement

4.3.5 128kbps

01
// initialise device with general reset
// this powers down everything excluding the xtal oscillator circuit

61 33
// setup analogue section - filters set for 7.0KHz mode

62 BE
// volume=0dB, sidetone= -21dB and off

63 80
// audio level=0dB

64 55
65 55
// power control, everything on low

70 00
// codec mode=ADM unbuffered (continuous bit serial mode)

71 0000
// data scrambler and descrambler both off

72 EC00
// using 8.0MHz master clock **(NOTE: this is a different master clock frequency than those used for previous settings.)**
// filter clock prescaler divider=4, filter clock main divider=8.0 => 256kHz internal SCF clock

// bit clock prescaler divider=1, encode and decode bit clock dividers=1, since constant divider=64 => 128kbps sampling rate

73 0078
// PLL is off, bypass PLL data filter and power it down
// internal decode and encode clocks from decoder internal clock (synchronised to XTAL/CLK input)

81 88
// enable encoder and decoder with no IRQs

D0 00B8
// setup decoder
// decimate by 8
// decode ADM input from RX Data
// ADM estimator drives output
// VAD attack tc=4ms and decay tc=128ms
// normal VAD outputs

D1 C9F9

```
E1      C9FC
        // ADM mode syllabic tc=10.7ms
        // step size dynamic range 5120/20
        // companding rule = 4 of 4
        // principal fc=318Hz
        // second order fc=2546Hz
        // encoder zero fc=8.1KHz decoder zero fc=N/A
        // decoder zero at 64kHz i.e. bit_rate/2 enabled

D2      0200
E2      0200
        // VAD thresholds ~20mv

D8      AA
E8      AA
        // prime idle pattern into C-BUS ADM input registers

E0      00B8
        // ADM encode feedback from comparator, otherwise as decoder

E3      0000
        // clear encoder offset input reg to disable idle channel enhancement
```

5 Conclusion

The CMX649 provides the designer with many options for voice coding. It is hoped that the settings provided in this document will assist the designer in rapidly selecting the best configuration for their application

CML Microcircuits (UK) Ltd COMMUNICATION SEMICONDUCTORS	CML Microcircuits (USA) Inc. COMMUNICATION SEMICONDUCTORS	CML Microcircuits (Singapore) Pte Ltd COMMUNICATION SEMICONDUCTORS
Tel: +44 (0)1621 875500 **Fax:** +44 (0)1621 875600 **Sales:** sales@cmlmicro.com **Tech Support:** techsupport@cmlmicro.com	**Tel:** +1 336 744 5050 800 638 5577 **Fax:** +1 336 744 5054 **Sales:** us.sales@cmlmicro.com **Tech Support:** us.techsupport@cmlmicro.com	**Tel:** +65 62 888129 **Fax:** +65 62 888230 **Sales:** sg.sales@cmlmicro.com **Tech Support:** sg.techsupport@cmlmicro.com

- www.cmlmicro.com -