


Communication System Design Using DSP Algorithms

**with Laboratory Experiments
for the TMS320C6713TM DSK**

Steven A. Tretter
*University of Maryland
College Park, MD*

 Springer

Contents

1	Overview of the Hardware and Software Tools	1
1.1	Some DSP Chip History and Typical Applications	2
1.2	The TMS320C6713 Floating-Point DSP	6
1.2.1	The 'C6000 Central Processing Unit (CPU)	6
1.2.2	Memory Organization for the TMS320C6713 DSK	11
1.2.3	Enhanced Direct Memory Access Controller (EDMA)	11
1.2.4	Serial Ports	12
1.2.5	Other Internal Peripherals	13
1.2.6	Brief Description of the TMS320C6000 Instruction Set	13
1.2.7	Parallel Operations and Pipelining	16
1.3	The TMS320C6713 DSP Starter Kit (DSK)	18
1.3.1	The Audio Interface Onboard the TMS320C6713 DSK	20
1.4	Software Support for the DSK Board and 'C6x DSP's	21
1.4.1	The Board Support Library (BSL)	21
1.4.2	The Chip Support Library	22
1.5	Code Composer Studio	22
1.5.1	Project Files and Building Programs	22
1.5.2	The Optimizing Compiler and Assembler	23
1.5.3	The Linker	25
1.5.4	Building Programs from Command Line Prompts	25
1.5.5	The Archiver	26
1.5.6	Additional Code Composer Studio Features	26
1.6	Other Software	27
1.6.1	Digital Filter Design Programs	27
1.6.2	Commercial Software	27
1.7	Introductory Experiments	28
2	Learning to Use the Hardware and Software Tools	29
2.1	Getting Started with a Simple Audio Loop Through Program	29
2.1.1	A Linker Command File and Beginning C Program	29
2.1.2	Properties of the AIC23 Codec	35
2.1.3	Creating a CCS Project for <code>dkstart32.c</code>	36
2.1.4	Experiment 2.1: Building and Testing <code>dkstart32.c</code>	37
2.2	More Details on the McBSP Serial Ports and Codecs	38

2.2.1	Basic McBSP Transmitter and Receiver Operation	38
2.2.2	Example C Code for Reading from and Writing to the Codec	41
2.3	The 'C6000 Timers	42
2.4	Generating a Sine Wave by Polling XRDY	43
2.4.1	Experiment 2.2: Instructions for the Polling Experiment	45
2.5	Generating a Sine Wave Using Interrupts	46
2.5.1	The CPU Interrupt Priorities and Sources	46
2.5.2	Interrupt Control Registers	46
2.5.3	What Happens When an Interrupt Occurs	50
2.5.4	TI Extensions to Standard C Interrupt Service Routines	51
2.5.5	Using the dsk6713bsl32 Library for Interrupts	51
2.5.6	Experiment 2.3: Generating Sine Waves by Using Interrupts	53
2.6	Generating a Sine Wave with the EDMA and a Table	56
2.6.1	EDMA Overview	56
2.6.2	EDMA Event Selection	57
2.6.3	Registers for Event Processing	58
2.6.4	The Parameter RAM (PaRAM)	59
2.6.5	Synchronization of EDMA Transfers	60
2.6.6	Linking and Chaining EDMA Transfers	61
2.6.7	EDMA Interrupts to the CPU	62
2.6.8	Experiment 2.4: Generating a Sine Wave Using the EDMA Controller	62
3	Digital Filters	67
3.1	Discrete-Time Convolution and Frequency Responses	67
3.2	Finite Duration Impulse Response (FIR) Filters	68
3.2.1	Block Diagram for Most Common Realization	68
3.2.2	Two Methods for Finding the Filter Coefficients to Achieve a Desired Frequency Response	69
3.3	Using Circular Buffers to Implement FIR Filters in C	72
3.4	Circular Buffers Using the 'C6000 Hardware	75
3.4.1	How the Circular Buffer is Implemented	75
3.4.2	Indirect Addressing Through Registers	76
3.5	Interfacing C and Assembly Functions	76
3.5.1	Responsibilities of the Calling and Called Function	76
3.5.2	Using Assembly Functions with C	79
3.6	Linear Assembly Code and the Assembly Optimizer	79
3.6.1	A Linear Assembly Convolution Function that Uses a Circular Buffer and Can be Called from C	81
3.7	Infinite Duration Impulse Response (IIR) Filters	89
3.7.1	Realizations for IIR Filters	89
3.7.2	A Program for Designing IIR Filters	92
3.7.3	Two Methods for Measuring a Phase Response	95
3.8	Laboratory Experiments for Digital Filters	96
3.8.1	Experiment 3.1: FIR Filters Entirely in C	96

3.8.2 Experiment 3.2: FIR Filters Using C and Assembly 97

3.8.3 Experiment 3.3: Implementing an IIR Filter 98

3.9 Additional References 98

4 The FFT and Power Spectrum Estimation 101

4.1 The Discrete-Time Fourier Transform 101

4.2 Data Window Functions 102

4.3 The Discrete Fourier Transform and its Inverse 104

4.4 The Fast Fourier Transform 104

4.5 Using the FFT to Estimate a Power Spectrum 112

4.6 Laboratory Experiments 113

4.6.1 Experiment 4.1: FFT Experiments 113

4.6.2 Experiment 4.2: Making a Spectrum Analyzer 114

4.7 Additional References 118

5 Amplitude Modulation 121

5.1 Theoretical Description of Amplitude Modulation 121

5.1.1 Mathematical Formula for an AM Signal 121

5.1.2 Example for Single Tone Modulation 122

5.1.3 The Spectrum of an AM Signal 123

5.2 Demodulating an AM Signal by Envelope Detection 123

5.2.1 Square-Law Demodulation of AM Signals 124

5.2.2 Hilbert Transforms and the Complex Envelope 125

5.3 Laboratory Experiments for AM Modulation and Demodulation 127

5.3.1 Experiment 5.1: Making an AM Modulator 128

5.3.2 How to Capture DSK Output Samples with CCS for Plotting 129

5.3.3 Experiment 5.2: Making a Square-Law Envelope Detector 130

5.3.4 Experiment 5.3: Making an Envelope Detector Using the Hilbert Transform 131

5.4 Additional References 132

6 DSBSC Amplitude Modulation and Coherent Detection 133

6.1 Mathematical Form for a DSBSC-AM Signal 133

6.2 The Ideal Coherent Receiver 134

6.3 The Costas Loop as a Practical Approach to Coherent Demodulation 136

6.4 Exercises and Experiments for the Costas Loop 138

6.4.1 Theoretical Design Exercises 139

6.4.2 Hardware Experiments 140

6.5 Additional References 141

7 Single-Sideband Modulation and Frequency Translation 143

7.1 Single-Sideband Modulators 143

7.2 Coherent Demodulation of SSB Signals 145

7.3 Frequency Translation 146

7.4 Laboratory Experiments 147

7.4.1	Experiment 7.1: Making an SSB Modulator	148
7.4.2	Experiment 7.2: Coherent Demodulation of an SSB Signal	148
7.5	Additional References	150
8	Frequency Modulation	151
8.1	The FM Signal and Some of its Properties	151
8.1.1	Definition of Instantaneous Frequency and the FM Signal	151
8.1.2	Single Tone FM Modulation	152
8.1.3	Narrow Band FM Modulation	154
8.1.4	The Bandwidth of an FM Signal	154
8.2	FM Demodulation by a Frequency Discriminator	154
8.2.1	An FM Discriminator Using the Pre-Envelope	155
8.2.2	A Discriminator Using the Complex Envelope	156
8.3	Using a Phase-Locked Loop for FM Demodulation	157
8.4	Laboratory Experiments for Frequency Modulation	160
8.4.1	Experiment 8.1: Measuring the Spectrum of an FM Signal	160
8.4.2	Experiment 8.2: FM Demodulation Using a Frequency Discriminator	161
8.4.3	Experiment 8.3: Using a Phase-Locked Loop for FM Demodulation	161
8.5	Additional References	162
9	Pseudo-Random Binary Sequences and Data Scramblers	163
9.1	Using Shift Registers to Generate Pseudo-Random Binary Sequences	164
9.1.1	The Linear Feedback Shift Register Sequence Generator	164
9.1.2	The Connection Polynomial and Sequence Period	165
9.1.3	Properties of Maximal Length Sequences	166
9.2	Self Synchronizing Data Scramblers	167
9.2.1	The Scrambler	167
9.2.2	The Descrambler	169
9.3	Theoretical and Simulation Exercises	169
9.3.1	Exercises for a Shift Register Sequence Generator with a Primitive Connection Polynomial	169
9.3.2	Exercises for a Shift Register Sequence Generator with an Irreducible but not Primitive Connection Polynomial	170
9.3.3	Exercises for a Shift Register Sequence Generator with a Reducible Connection Polynomial	171
9.4	Additional References	171
10	The RS-232C Protocol and a Bit-Error Rate Tester	173
10.1	The EIA RS-232C Serial Interface Protocol	173
10.2	Error Rate for Binary Signaling on the Gaussian Noise Channel	176
10.3	The Navtel Datatest 3 Bit Error Rate Tester	177
10.4	Bit-Error Rate Test Experiment	178
10.5	Additional References	185

11 Digital Data Transmission by Pulse Amplitude Modulation	187
11.1 Description of a Baseband Pulse Amplitude Modulation System	187
11.2 Baseband Shaping and Intersymbol Interference	190
11.2.1 The Nyquist Criterion for No ISI	190
11.2.2 Raised Cosine Baseband Shaping Filters	191
11.2.3 Splitting the Shaping Between the Transmit and Receive Filters	192
11.2.4 Eye Diagrams	192
11.3 Implementing the Transmit Filter by an Interpolation Filter Bank	194
11.4 Symbol Error Probability with Additive Gaussian Noise	194
11.5 Symbol Clock Recovery	196
11.6 Simulation and Theoretical Exercises for PAM	198
11.6.1 Generating Four-Level Pseudo-Random PAM Symbols	198
11.6.2 Eye Diagram for a PAM Signal Using a Raised Cosine Shaping Filter	199
11.6.3 Eye Diagram for a PAM Signal Using a Square-Root of Raised Cosine Shaping Filter	199
11.6.4 Theoretical Error Probability for a PAM System	200
11.7 Hardware Exercises for PAM	200
11.7.1 Generating a PAM Signal and Eye Diagram	200
11.7.2 Testing the Square-Law Symbol Clock Frequency Generator	201
11.7.3 Optional Team Exercise	202
11.8 Additional References	203
12 Variable Phase Interpolation	205
12.1 Continuously Variable Phase Interpolation	205
12.1.1 Computing the Least-Squares Fits	208
12.2 Quantized Variable Phase Interpolation	208
12.3 Closing the Tracking Loop	209
12.4 Changing the Sampling Rate by a Rational Factor	211
12.5 Experiments for Variable Phase Interpolation	213
12.5.1 Experiment 12.1: Open Loop Phase Shifting Experiments	213
12.5.2 Experiment 12.2: Making a Symbol Clock Tracking Loop	213
12.6 Additional References	214
13 Fundamentals of Quadrature Amplitude Modulation	215
13.1 A Basic QAM Transmitter	215
13.2 Two Constellation Examples	217
13.2.1 The 4×4 16-Point Constellation	218
13.2.2 A 4-Point Four Phase Constellation	220
13.3 A Modulator Structure Using Passband Shaping Filters	221
13.4 Ideal QAM Demodulation	223
13.5 QAM Modulator Experiments	224
13.5.1 Steps to Follow in Making a Transmitter	225
13.5.2 Testing Your Transmitter	226
13.5.3 Generating a Startup Sequence	227

13.6	Additional References	228
14	QAM Receiver I – Clock Recovery and Other Front-End Subsystems	229
14.1	Overview of a QAM Receiver	229
14.2	Details About the Receiver Front-End Subsystems	231
14.2.1	Automatic Gain Control	231
14.2.2	The Carrier Detect Subsystem	232
14.2.3	Symbol Clock Recovery	232
14.3	Experiments for the QAM Receiver Front-End	239
14.4	Additional References	240
15	QAM Receiver II – Equalizer and Carrier Recovery System	241
15.1	The Complex Cross-Coupled Passband Adaptive Equalizer	241
15.1.1	The LMS Method for Adjusting the Equalizer Tap Values	242
15.1.2	Theoretical Behavior of the LMS Algorithm	247
15.1.3	Adding Tap Leakage to the LMS Algorithm	248
15.2	The Phase-Splitting Fractionally Spaced Equalizer	249
15.3	Decision Directed Carrier Tracking	251
15.4	Blind Equalization	253
15.4.1	Blind Equalization with the Complex Cross-Coupled Equalizer	254
15.4.2	Blind Equalization with the Phase-Splitting Equalizer	255
15.5	Complex Cross-Coupled Equalizer and Carrier Tracking Experiments	256
15.5.1	Implementing the Slicer	256
15.5.2	Making a Demodulator and Carrier Tracking Loop	258
15.5.3	Making a Complex Cross-Coupled Adaptive Equalizer	259
15.5.4	Bit-Error Rate Test	259
15.5.5	Optional Experiment – Receiving the 16-Point V.22bis Constellation	259
15.5.6	Optional Experiment – Ideal Reference Training	260
15.6	Optional Phase-Splitting Fractionally Spaced Equalizer Experiment	260
15.7	Optional Blind Equalization Experiment	261
15.8	Additional References	262
16	Echo Cancellation for Full-Duplex Modems	263
16.1	The Echo Sources in a Dialed Telephone Line Circuit	263
16.2	The Data-Driven, Nyquist, In-Band Echo Canceler	265
16.2.1	General Description	265
16.2.2	The Near-End Echo Canceler	267
16.2.3	The Far-End Echo Canceler	269
16.2.4	Far-End Frequency Offset Compensation	270
16.3	Echo Canceler Experiments	271
16.3.1	Making a Near-End Echo Canceler	271
16.3.2	Making a Far-End Echo Canceler with Frequency Offset Correction	271
16.4	Additional References	272

17 Multi-Carrier Modulation	273
17.1 History and Implementation of Multi-Carrier Modulation	273
17.2 Asymmetric Digital Subscriber Line (ADSL) System Architecture	277
17.3 Components of a Simplified ADSL Transmitter	278
17.3.1 The Cyclic Redundancy Check Generator	278
17.3.2 The Scrambler	281
17.3.3 The Reed-Solomon Encoder	281
17.3.4 The Convolutional Interleaver	282
17.3.5 The Map and IFFT Modulator Blocks	285
17.3.6 Some Signals Used for Initialization and Synchronization	288
17.4 A Simplified ADSL Receiver	290
17.4.1 Demodulation and Frequency Domain Equalization	291
17.4.2 Sample Clock Acquisition and Tracking	292
17.4.3 Symbol Alignment Acquisition and Tracking	296
17.4.4 Remaining Blocks	297
17.5 Making a Simplified ADSL Transmitter and Receiver	298
17.5.1 Making a 64-Point IFFT and a 64-Point FFT	298
17.5.2 Making a Scrambler, Constellation Point Mapper, and Their Inverses	299
17.5.3 Measuring the Channel Impulse Response Duration	299
17.5.4 Completing the Transmitter	300
17.5.5 Making the Receiver	301
17.6 Additional References	304
18 Suggestions for Additional Experiments	307
18.1 Elementary Modem Handshake Sequence	307
18.2 Make an ITU-T V.21 Frequency Shift Keyed (FSK) Modem	307
18.3 Fast Equalizer Training Using Periodic Sequences	308
18.4 Trellis Coded Modulation	309
18.5 Reed-Solomon Encoder and Decoder	309
18.6 Turbo Codes	309
18.7 Low Density Parity Check Codes	310
18.8 V.34 Constellation Shaping by Shell Mapping	310
18.9 Nonlinear Precoding for V.34	311
18.10 Speech Coders	311
A Generating Gaussian Random Numbers	313
A.1 The 'C6713 C Compiler Pseudo Random Number Generator	313
A.2 A Better Uniform Random Number Generator	314
A.3 Turning Uniformly Distributed Random Variables into a Pair of Gaussian Random Variables	316
A.4 Limit on the Peak of the Simulated Gaussian Random Variables	317
B A TTL/RS-232C Interface for McBSP0	319
C Equipment List for Each Station	323

References	325
I. List of Manuals	325
II. Selected Reference Books and Papers	325
A. DSP Laboratory Books Using DSP Hardware	325
B. DSP Laboratory Books Using Software Simulation	326
C. Books and Papers on Digital Signal Processing	327
D. Books and Papers on Communications	328
E. References for Wireline and Wireless Multi-Carrier Modulation	331
F. Books and Papers on Error Correcting Codes	332
III. Interesting Web Sites	333
Index	335