

EE442/EE592 Real-Time Digital Signal Processing

Project #4: TI DSP-Based BPSK Modem

Due: 5:00pm, Friday, April 15

Assignment

The goal of this project is build a BPSK modem consisting of a transmitter code and a receiver code (each residing on a separate DSK) capable of transmitting data (simplex connection) over a standard audio cable. The complete theory and algorithm of the project is described in the textbook. The modem will use the parameters discussed in lecture: data rate, $R_b = 250$ bps; carrier frequency, $f_c = 2000$ Hz; $f_s = 8000$ Hz; and 7 bit ASCII characters which transmitted in a frame composed of a start bit ('0'), 7 data bits, parity bit, and two stop bits ('1 1'). Please use even parity for this project. The waveform is to be digitally synthesized using **lookup_waveIntDelta.c** with the data bits modulating the phase of the carrier ('0' induces a 0 radian phase, '1' induces a π radian phase).

MATLAB Simulation and user_data.h

Before you begin this project, it is strongly advised that you review a MATLAB simulation (**bpsk_modem.m**) of the data transmission/reception process to verify your understanding of the algorithm. In addition, a `user_data.h` file is also available. Both files can be downloaded from the class website

http://www.ece.nmsu.edu/~pdeleon/EE592/TI_6416_Code.html

Code Development

Since there are a number of routines that need to be written for the BPSK modem you should break down the project into small pieces identifying the major routines for the transmitter and those for the receiver. Begin the project by coding the transmitter. During transmitter development, you may wish to capture the transmitted waveform using the sound card and verify the phase changes corresponding to bit changes in MATLAB. You can also use this captured waveform and playback to test the receiver.

Algorithm

Complete algorithms for the transmitter and receiver are given in the textbook. Note that you do not need to verify parity upon reception.

Testing

We will test the performance of the modem using two separate TI DSK boards—one which runs the transmitter code and the other which runs the receiver code. An audio cable is strung between the two boards connecting the output of the transmitter with the input of the receiver. Both codes are executed and then the receiver is halted shortly afterward to verify correct ASCII values in memory. Evaluators will choose several random ASCII characters for transmission and you must demonstrate that these ASCII characters appear in your receiver's memory.

Submitted Items

You will be required to submit several items for this project, described below.

Code Printout

Please turn in printouts of your code including, **user_data.h**, **util.h**, **initialize_program.c**, **process_signal.c**, and any new files you write. Your codes should be fully documented in the header and completely commented.

Code in Electronic Form

Turn in a USB flash drive (with your name clearly labeled on it) with two directories: TX contains the transmitter code and RX contains receiver code. Each directory should contain **ALL** code and support files necessary to build the program.

Grading

We will evaluate, test, and grade the codes after the due date. A working code which correctly transmits and

receives data will yield +95 points (EE592); incorrect transmission will scale the point total accordingly. Students in EE442 please read on for grading information. In addition, good code design, readability, and commenting will be evaluated and scored up to an additional +5 points. If we have questions or problems with the project, we will contact you.

Bonuses

EE442

Since it is hard to break the modem into an EE442 project and an EE592 project, EE442 will be scored differently. For students enrolled in EE442, successful coding of the transmitter will yield +70 points and successful coding of the receiver will yield an additional +70 points. A successful transmitter but an incomplete receiver will yield a score between +70 and +140 points. It is expected that code is well-commented.

Maximum grade for EE442, Project #4 is 140 points.

EE592

Code your modem so that if BPS (in user_data.h) is changed to either 250, 500, 1000, or 2000, the modem will transmit and receive at that rate. No other parameters or code should have to be changed to make this work (+15 points).

Maximum grade for EE592, Project #4 is 115 points.

Evaluation

Setup/Testing

Transmitter only. Test procedure:

- 1) Capture the waveform corresponding to a random ASCII character.
- 2) Examine waveform in MATLAB for phase changes. You may use Chad Lackey's tools.
- 3) Verify parity bit for at least two ASCII codes with different number of ones

Transmitter/Receiver. Test procedure:

- 1) Transmit ASCII codes for student's name
- 2) Examine receiver memory for these characters.
- 3) Verify parity bit for at least two ASCII codes with different number of ones.

If Transmitter/Receiver does not function properly, it is the student's responsibility to demonstrate what is working. Suggestions:

- 1) Show pilot tone is emitted.
- 2) Show correct matched filtering.
- 3) Show first start bit is transmitted (audible click)
- 4) Show data bits are transmitted (must show for each sample we return to top of main event loop).
- 5) Show parity/stop bits are transmitted
- 6) Show first frame is sent (MATLAB waveform analysis)

EE442

Baseline: Operational transmitter +65

Commenting/Code structure: +5

Increments to baseline:

- +10 Basic receiver code written but not functioning
- +20 Advanced receiver code written but not functioning
- +40 Receiver synchronizes with transmitter
- +60 Receiver demodulates single frame
- +70 Operational receiver

EE592

Baseline: Correct data transmission +95

Commenting/Code structure: +5

Partials:

+47 Operational transmitter

+48 Operational receiver

All routines written but not working +12

Synchronization +12

Single frame demodulated +12

All frames demodulated +12