GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING

ECE 4271 SPRING 2012
LabVIEW and Matlab PROJECT #3

# *Wireless DTMF Transceiver Design Using USRP*

Assigned: Tuesday, April 10, 2012
**Due: Tuesday, April 24, 2012 4:30pm**

- This project is to be done *teamly.* Any project-specific help needed should be sought from Dr. Ma. Students are not to discuss the theory or approaches to coding the theory with one another, nor are they to assist in debugging each other's work. ***The Georgia Tech honor code and its policies apply.***
  You may ask Dr. Ma or the TA questions regarding theory and implementation of the project, including asking them at the beginning or end of class or during office hours, when others can benefit as well.
- Reports and code will be graded primarily on completeness in addressing the assignment and quality of results. Reports must be typed, not handwritten; should be well-organized; and must clearly explain your design and the decisions and analysis that went into it. Code must be provided in the form specified below, and should be commented well-enough to be clearly understandable.
- Questions or clarifications should be directed to Dr. Ma.[1] Errata, revisions and hints (if any) will be made available via e-mail, the class T-square site, or during class.

[1] Office: 404-385-7456. Office hours: Tuesdays 3:00 – 4:30 PM, or by appointment, else e-mail: <xiaoli@ece.gatech.edu>.

## 1. PROBLEM

We use NI LabVIEW and Matlab to simulate a digital communication system for DTMF signals. DTMF encoders and decoders will be using Matlab and basic digital communication transmitter and receiver through USRP will be designed by LabVIEW.

## 2. REQUIREMENTS

You must submit a report of your methods and findings that must include:

1. An overview description of how your algorithm works. This must include a block diagram of the major processing steps with a description of the rationale for each step. It should identify the major LabVIEW/Matlab functions or code blocks corresponding to each major function in your block diagram.

2. A brief description of the theory underlying the processing algorithm. Explain the analyses and design decisions that led to each of your parameter choices.

3. A complete listing of the LabVIEW program used to process the signals. Code must be sufficiently modular and well-arranged to be understood readily. Listings must also be included for any functions used, except for those which are built into either LabVIEW or its Modulation Toolkit, or were provided in class.

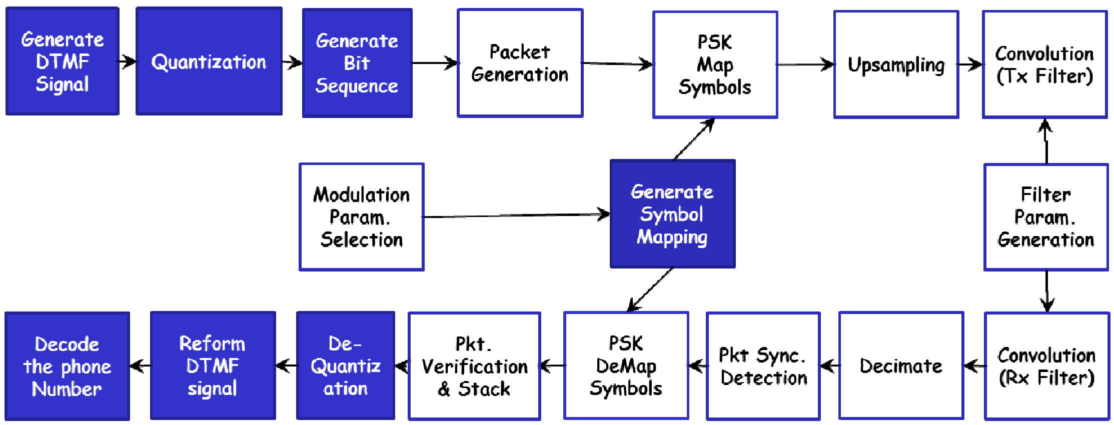
In addition to the written report, you must submit your LabVIEW VI files to Dr. Ma. E-mail a single `.zip` file containing multiple LabVIEW VIs to xiaoli@ece.gatech.edu ***prior to the deadline***. See below for further instructions on naming the zip file and how your code must be organized.

***Start working early. If you encounter problems, please ask questions to clarify anything that is vague or incomplete.***

## 3. MAJOR BLOCKS

3.1 Overall block diagram structure

The figure shows the basic block structure and the corresponding VI files of wireless DTMF transceiver.




You should implement the **blue-shaded blocks** while the white blocks are offered by default. You can also modify white blocks to enhance the performance while keeping the main packet frame structure.

Basically the program mainly consists of 3 parts; digitalized DTMF encoder, USRP digital communication transceiver, digitalized DTMF decoder. You will mainly design digitalized DTMF encoder/decoder parts, but it is also available to modify USRP digital communication transceiver part as well in order to improve the performance.

3.2 Digitalized DTMF encoder

Given the phone number string, only phone numbers are extracted and converted to numeric array. From this array, generate the DTMF signal using DTMF encoder. Next, quantize and encode this signal into input bit streams. You should design the quantization and source encoding block using the given rule.

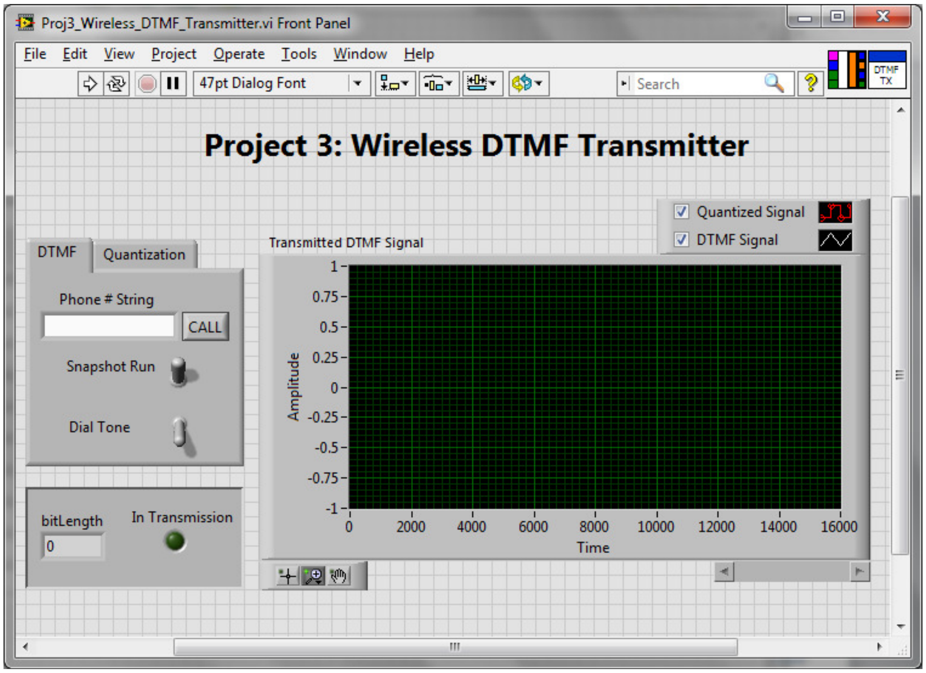
3.2.1 VI file hierarchy

Implement and modify **the bold VIs** in the 'subVIs_DTMF' folder

- Proj3_Wireless_DTMF_Transmitter.vi

- sub_DTMF_StringToNumber.vi
- **sub_DTMF_Encoder.vi**
- **sub_DTMF_Quantize_BitEncoding.vi**

Note. You should also open the other VIs in order to figure out how to operate.




3.3 USRP transceiver

Use the default USRP transceiver. You can use your own symbol mapping vi files. Generally, the procedure of the USRP transceiver is as follows:

1. The generated bit streams are divided into multiple segments.
2. The header and footer information is appended to each segment in order to make one packet.
3. Generated packets are transmitted repeatedly from the transmitter.
4. The signals are captured from the air at the receiver.
5. The receiver decodes the header information of the packet and extracts the segment part from the packet.

6. The extracted segments are saved in the memory, and if all the packets are received successfully, the saved segments are assembled together to make the output bit streams.

### 3.4 Digitalized DTMF Decoder

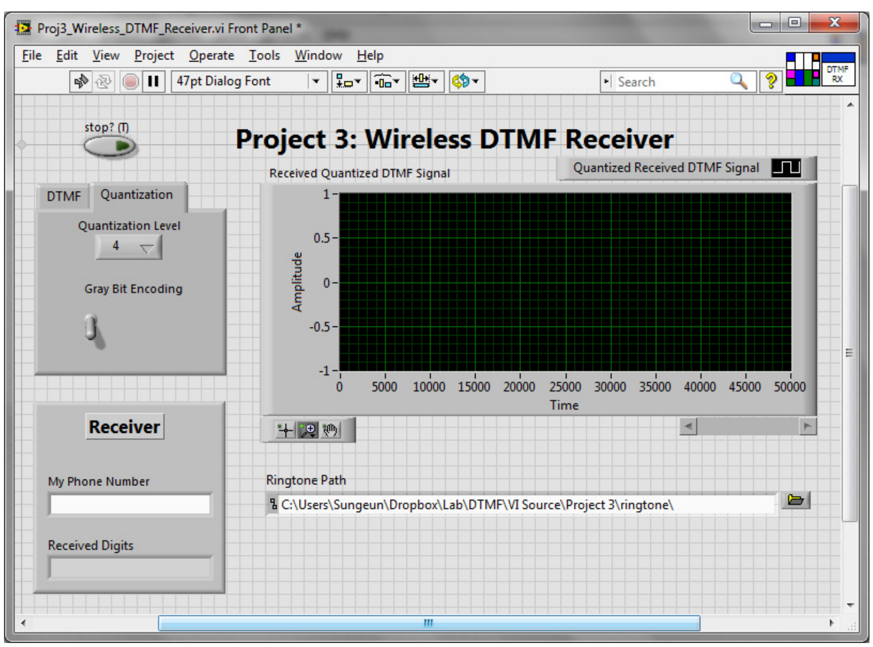
Given the output bit streams, de-quantize the stream and reconstruct the DTMF signal. Then, using DTMF decoder, get the phone number. You should design the de-quantization and source decoding block using the given rule. Before the de-quantization, bit padding will be performed in order to make the length of the bitstream as the multiple of the quantization bits. Here is the example how bitpadding block should be implemented when the quantization level *L=8*.

| Received bitstream | Padded bits *(L=8)* | Output bitstream |
|---|---|---|
| **010**010**101**0 | 00 | **010**010**101**000 |

#### 3.4.1 VI file hierarchy

Implement and modify **the bold VIs** in the ‘subVIs_DTMF’ folder

- Proj3_Wireless_DTMF_Receiver.vi
  - **sub_DTMF_BitPadding.vi**
  - **sub_DTMF_Dequantize_BitDecoding.vi**
  - **sub_DTMF_Decoder.vi**



3.5 Quantization and Source Encoding/Decoding Rule

3.5.1 Quantization Rule

- Quantization Level: $L$
- Quantization range: $x_{\max} = 1$, $x_{\min} = -1$
- Quantization size: $Q = \frac{x_{\max} - x_{\min}}{L}$
- Origianal input signal: $x$
- Quantization function $f(x) = \begin{cases} x_{min}, & x < x_{min} \\ x, & x_{min} \leq x < x_{max} \\ x_{max-} \frac{Q}{2}, & x_{max} \leq x \end{cases}$
- Quantizated output signal: $x_{\text{quant}} = \left\lfloor \frac{f(x)}{Q} \right\rfloor \cdot Q + \frac{Q}{2}$
- Quantized output level index: $l = \left\lfloor \frac{f(x)}{Q} \right\rfloor + \left\lfloor \frac{x_{min}}{Q} \right\rfloor$

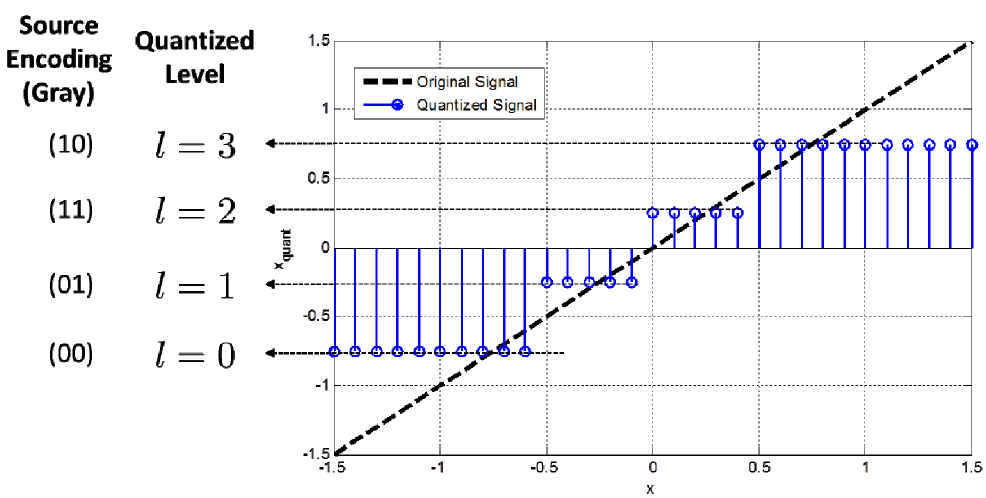



Figure 1 Quantization example, ***L*=4, $x_{\min}$ = -1, $x_{\max}$ = 1**

You should design the dequantization block as well based on the quantization rule.

### 3.5.2 Source Encoding/Decoding Table

- 2 Level (*L=2*) case

| Level (*l*) | 0 | 1 |
|---|---|---|
| Typical | 0 | 1 |
| Gray | 0 | 1 |

- 4 Level (*L=4*) case

| Level (*l*) | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| Typical | 00 | 01 | 10 | 11 |
| Gray | 00 | 01 | 11 | 10 |

- 8 Level (*L=8*) case

| Level (*l*) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Typical | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| Gray | 000 | 001 | 011 | 010 | 110 | 111 | 101 | 100 |

- 16 Level (*L=16*) case

| Level (*l*) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Typical | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| Gray | 0000 | 0001 | 0011 | 0010 | 0110 | 0111 | 0101 | 0100 |
| Level (*l*) | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Typical | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| Gray | 1100 | 1101 | 1111 | 1110 | 1010 | 1011 | 1001 | 1000 |

### 3.6 DTMF Encoder/Decoder Test

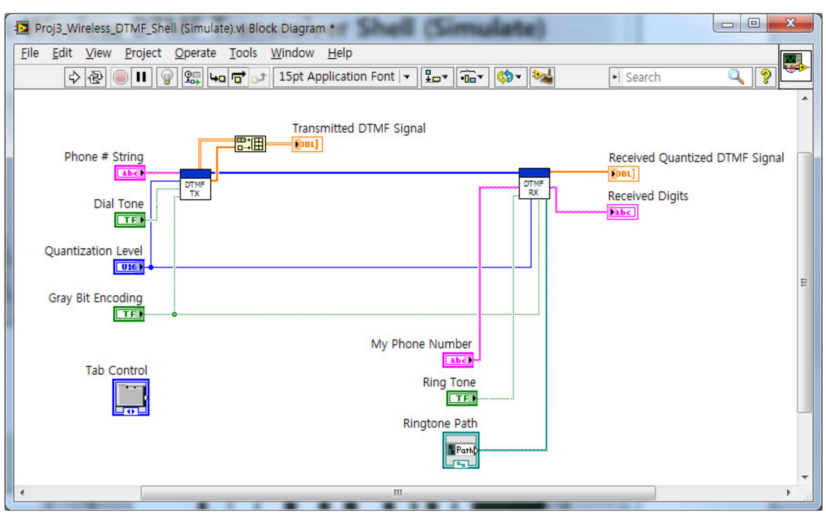



**Figure 2 Block diagram for the offline DTMF encoder/decoder test**

You can test and verify your DTMF encoder/decoder blocks only (offline test without USRP transceiver) using the following vi in the 'subVIs_DTMF' folder. Your encoder/decoder should be worked without any error on this offline test.

- **Proj3_Wireless_DTMF_Shell (Simulate).vi**

You will make the phone number string at the transmitter, and if run the VI, you should get the same phone number string at the receiver as shown in the figure.

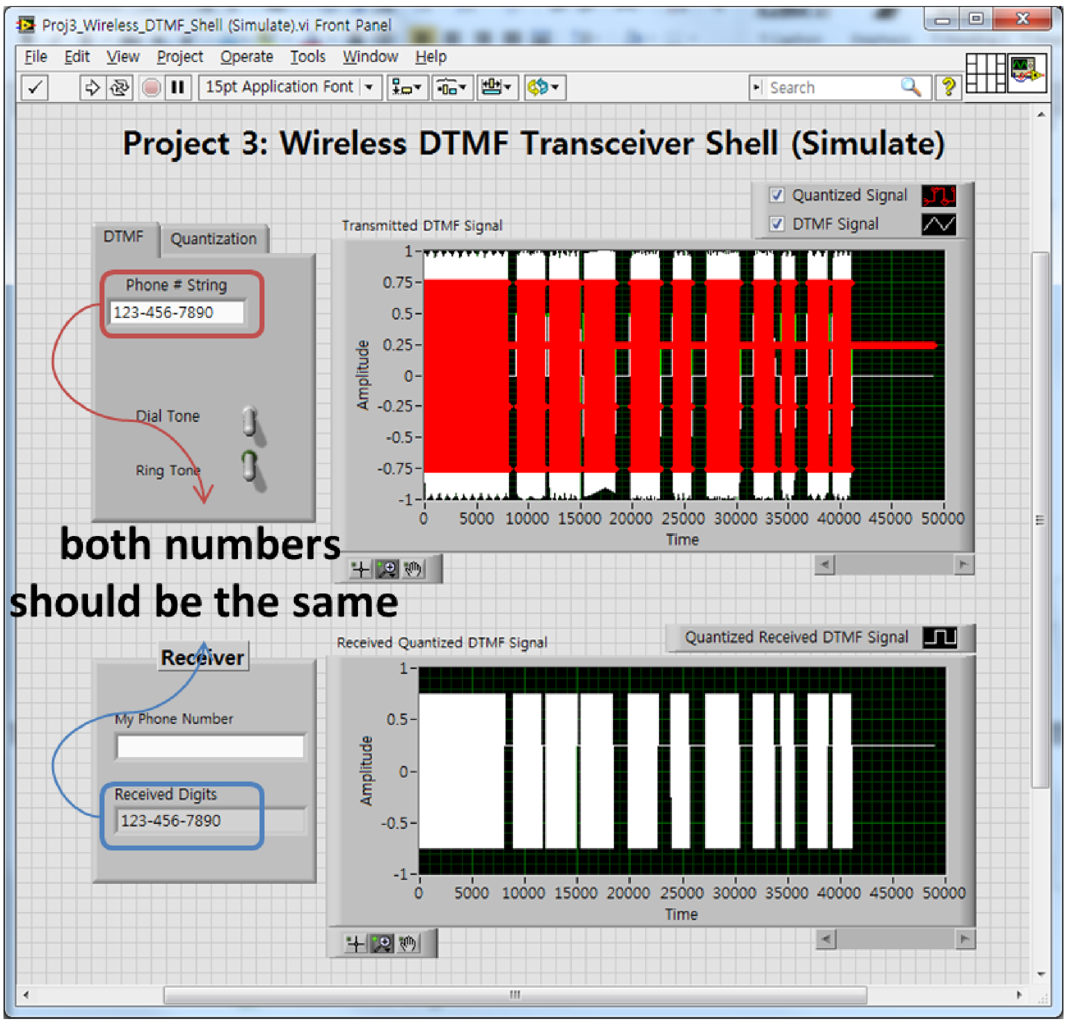



**Figure 3 Front panel of the DTMF Shell (Simulate), you should test whether your DTMF encoder/decoder works properly.**

## 4. Live Demo and COMPETITION

On April 19 4:35-5:55, we will ask each group to do a live demo and present the new features. Peer evaluation (from other teams) will be performed.

We will also have the student competition between multiple teams.

The instructor will give the system parameter setup, and will dial the tone and transmit the corresponding encoded DTMF signal over-the-air. All the teams will receive this encoded signal at the same time, and the very first team which decodes the DTMF signal and rings a bell will win the competition.

## 5. LAVIEW CODE SUBMISSION AND REPORT EVALUATION

Your LABVIEW code must be submitted by e-mail in a single zip file to xiaoli@ece.gatech.edu before the project deadline. The following conditions apply to this zip file and your code:

1. The file must contain all of the LABVIEW code, including all VIs and subVIs, data files, *etc.* required to run your algorithm on a test signal. Make all of these files with the name `FirstInitialLastName.zip`.

## 6. GRADING

In this part, we will use USRP devices to test wireless communications under different environments. Each team will have two GNU radios (one as transmitter and one as receiver). You have flexibility to design your tests to show the performance. The comparisons need to cover:

(1) Different constellation sizes (BPSK, QPSK, 8PSK).

(2) Different environments (open area with line of sight, crowded area without line of sight)

(3) Different interference levels (different groups run together, groups run individually)

In your report, you need to have:

(1) error-rate (number of reception errors over the total number of trials)

(2) raw received data in the complex plane with real and imaginary axes

(3) description or picture of each test case

(4) your conclusions

The grading for this project is:

- Codes 40%
    - o Work for different modulation sizes and quantization levels
- Report 40%
    - o Code listing and readability: 20%
    - o General report quality (complete but succinct, figures, writing, *etc.*): 20%
- Peer evaluation: 10%
- Live Demo and Competition: 10%
- New Features: 10% (bonus)