Connexions module: m11849

DIGITAL TRANSMITTER: INTRODUCTION TO FREQUENCY SHIFT KEYING*

Robert Morrison Matt Kleffner Michael Frutiger

Based on Digital Transmitter: Introduction to Frequency Shift Keying[†] by Matthew Berry

This work is produced by The Connexions Project and licensed under the Creative Commons Attribution License ‡

Abstract

This module introduces frequency shift keying (FSK) and describes components of an FSK transmitter block-diagram.

1 Frequency Shift Keying

Frequency Shift Keying (FSK) is a scheme to transmit digital information across an analog channel. Binary data bits are grouped into blocks of a fixed size, and each block is represented by a unique carrier frequency, called a **symbol**, to be sent across the channel. ¹ This requires having a unique symbol for each possible combination of data bits in a block. In this laboratory exercise each symbol represents a two-bit block; therefore, there will be four different symbols.

The carrier frequency is kept constant over some number of samples known as the symbol period (T_{symb}) . The symbol rate, defined as F_{symb} , is a fraction of the board's sampling rate, F_s . For our sampling rate of 44.1 kHz and a symbol period of 32, the symbol rate is 44.1 k/32 symbols per second.

^{*}Version 1.3: Feb 24, 2004 10:40 pm -0600

[†]http://cnx.org/content/m10659/2.4/

[‡]http://creativecommons.org/licenses/by/1.0

¹The receiver then looks at the recovered symbol frequency to determine which block of bits was sent and converts it back to the appropriate binary data.

Connexions module: m11849 2

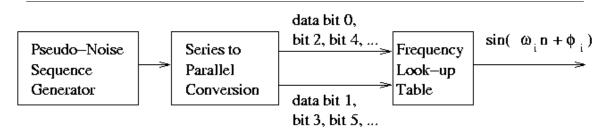


Figure 1: Pseudo-noise sequence generator and FSK transmitter.

2 Pseudo-Noise Sequence Generator

The input bits to the transmitter are provided by the special shift-register, called a **pseudo-noise sequence generator** (**PN generator**), on the left side of Figure 1. A PN generator produces a sequence of bits that appears random. The PN sequence will repeat with period $2^B - 1$, where B is the width in bits of the shift register. A more detailed diagram of the PN generator alone appears in Figure 2.

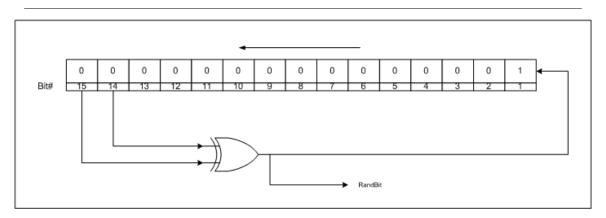


Figure 2: PN generator.

As shown in Figure 2, the PN generator is simply a shift-register and XOR gate. Bits 14 and 15 of the shift-register are XORed together and the result is shifted into the lowest bit of the register. This lowest bit is the output of the PN generator.

The PN generator is a useful source of random data bits for system testing. We can simulate the bit sequence that would be transmitted by a user as the random bits generated by the PN generator. Since communication systems tend to randomize the bits seen by the transmission scheme so that bandwidth can be efficiently utilized, the PN generator is a good data model.²

²PN generators have other applications in communications, notably in the Code Division Multiple Access schemes used by cellular telephones.

Connexions module: m11849

3 Series-to-Parallel Conversion

The shift-register produces one output bit at a time. Because each symbol the system transmits will encode two bits, we require the series-to-parallel conversion to group the output bits from the shift-register into blocks of two bits so that they can be mapped to a symbol.

4 Frequency Look-up Table

This is responsible for mapping blocks of bits to one of four frequencies as shown in Figure 1. Each possible two-bit block of data from the series-to-parallel conversion is mapped to a different carrier frequency ω_i

NOTE: Note that the subscript i denotes a symbol's index in the transmitted signal; i.e., the first symbol sent has index i = 1, the second symbol sent has index i = 2, and so on. Therefore, ω_i denotes the frequency and ϕ_i denotes the phase offset of the i^{th} transmitted symbol.

These frequencies are then used to generate the waveforms. The mappings for this assignment are given in Table 1.

Data Chunk	Carrier Frequency ω_i
00	$\frac{9\pi}{32}$
01	$\frac{13\pi}{32}$
11	$\frac{17\pi}{32}$
10	$\frac{21\pi}{32}$

Table 1

One way to implement this mapping is by using a look-up table. The two-bit data block can be interpreted as an offset into a frequency table where we have stored the possible transmission frequencies. Note that since each frequency mapping defines a symbol, this mapping is done at the symbol rate F_{symb} , or once for every T_{symb} DSP samples.

The symbol bit assignments are such that any two adjacent frequencies map to data blocks that differ by only one bit. This assignment is called **Gray coding** and helps reduce the number of bit errors made in the event of a received symbol error.

5 Phase Continuity

In order to minimize the bandwidth used by the transmitted signal, you should ensure that the phase of your transmitted waveform is continuous between symbols; i.e., the beginning phase of any symbol must be equal to the ending phase of the previous symbol. For instance, if a symbol of frequency $\frac{9\pi}{32}$ begins at phase 0, the symbol will end 31 output samples later at phase $31\frac{9\pi}{32}$. To preserve phase continuity, the next output sample must be at phase $32\frac{9\pi}{32}$, which is equivalent to phase π . Therefore, the next symbol, whatever its frequency, must begin at phase π . For each symbol, you must choose ϕ_i in the expression $\sin(\omega_i n + \phi_i)$ to create this continuity.