Electronics Communications Laboratory
Practical Session 4:
Digital Communications. ASK Transceiver

4.1 Introduction.

This practice proposes to implement an ASK transmitter and receiver using the DSP development board TMS320C6713. The binary stream source will be a function generator connected to the DSP through the speech-band Analog-to-Digital Converter (ADC). The output provided by the DSP and converted by the Digital-to-Analog Converter (DAC) will be analyzed in the oscilloscope.

The practice consists of 4 parts:
1) Introduction to the main elements of the practice
2) ASK transmitter design
3) ASK receive design
4) (Optional) DBPSK emitter and receiver design

4.2 Elements of the practice

To carry out this practice, the laboratory put at your disposal the following elements:
- 1 computer;
- 1 Texas Instruments TMS320C6713 DSP board
- 1 oscilloscope
- 1 function generator
- 2 connectors BNC/jack Stereo 3mm
- 1 connector BNC/BNC
- 1 connector jack/jack (Stereo)
- 1 adaptor BNC in "T"
-

4.2.1 Connection of the elements of the first part of the practice

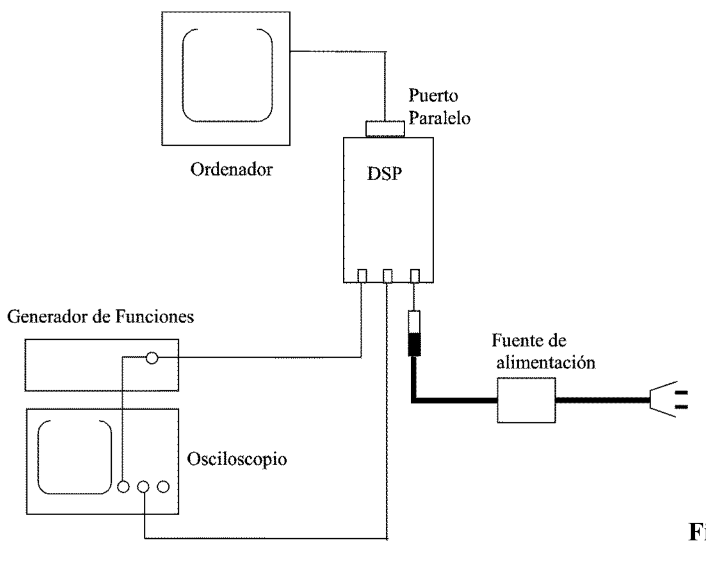


Figure 1: Connection for Part 1

For the first part of the practice, the elements should be connected as shown in Figure 1. The output of the DSP should be connected to the one of two channels of the oscilloscope using one of the two BNC/jack. The output of the function generator should be connected to the other channel of the oscilloscope and to the input of the DSP, using the BNC adaptor in “T”.

CAUTION: DO NOT CONNECT THE OUTPUT OF THE DSP TO THE OUTPUT OF THE FUNCTION GENERATOR

Note: The Jack/Jack connected is used only in the second part of the practice.

4.2.2 Connection of the elements for the second part of the practice

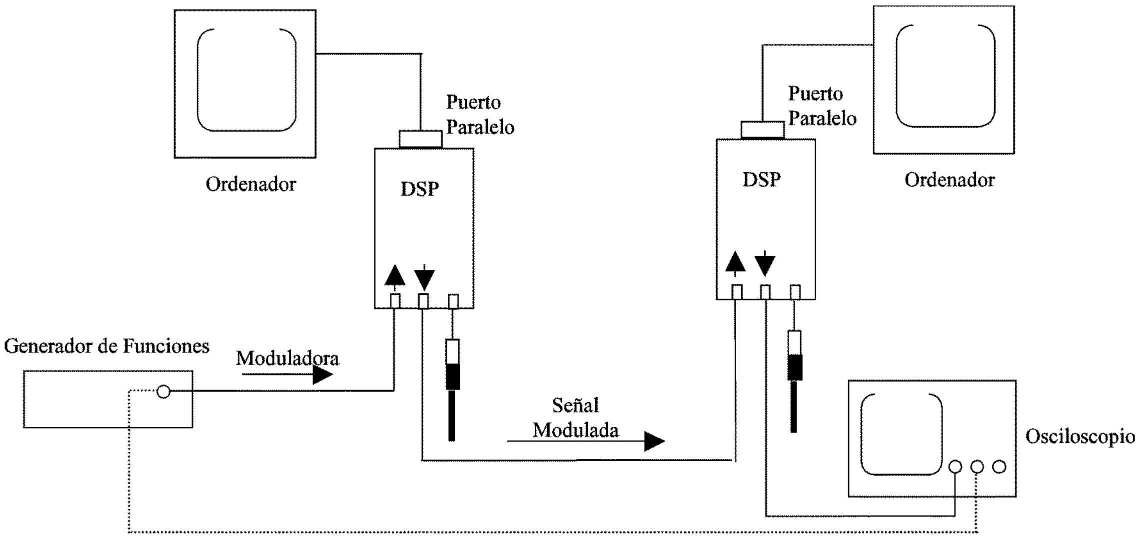


Figure 2: Connection for Part 2

The second part is carried out using two DSPs. Therefore, for this part, you need to re-organize in groups of 4. One of the DSP will be used to implement the program of the ASK emitter, while the other DSP will run the program corresponding to the ASK receiver.

The input of the transmitter DSP should be connected to the function generator. You should verify that the transmitter’s output is actually an ASK modulated. Then, the output of this DSP should be connected to the input of the other receiver DSP, using the cable jack/jack. The output of the DSP receiver should be connected to one of the oscilloscope’channels. It may be useful to connect the output of the function generator to the other channel of the oscilloscope to visualize the modulating signal.

CAUTION: DO NOT CONNECT THE OUTPUTS OF THE TWO DSPS. THIS MAY IRREVERSIBLY DAMAGE THEM.

4.3 Development of the practice

In the practices directory, you are provided with the obligatory part that should be included in all DSP codes, i.e., the initialization of the interrupt table, the codec programming, and other basic tasks required by the DSP to communicate with the external world.

This code is included in the two projects: **MOD_ASK.zip** and **DEMOD_ASK.zip** of the directory of the practices of this class.

4.4 Implementation of an ASK emitter

The first part of this practice consists in programming an ASK transmitter. The carrier frequency should be set to f_c = 10 kHz and the sampling frequency 48KHz (approx.). The output of the emitter should be $y(t) = \frac{1}{2}\{1 - sign(x)\}\cos 2\pi f_c t$, as highlighted by Figure 3. Moreover, the structure of the program should follow the three subsequent subsections.

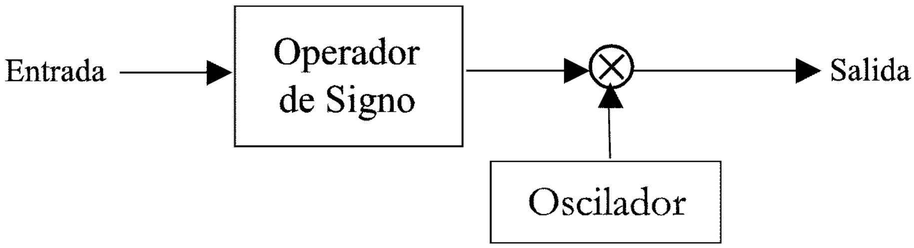


Figure 3: Structure of the ASK transmitter

4.4.1 Carrier generation using an unstable filter

There are two approaches to generate sinusoidal signals using DSPs. The first technique, already seen in the previous practice, uses a Look-Up Table (LUT), where samples of the sinusoidal signal are stored and reproduced. The second approach consists in the design of a digital oscillator based on the unstable 2nd order IIR filter (its z-transform has poles on the unit circle) depicted in the Figure 4, and which will be analyzed in this section.

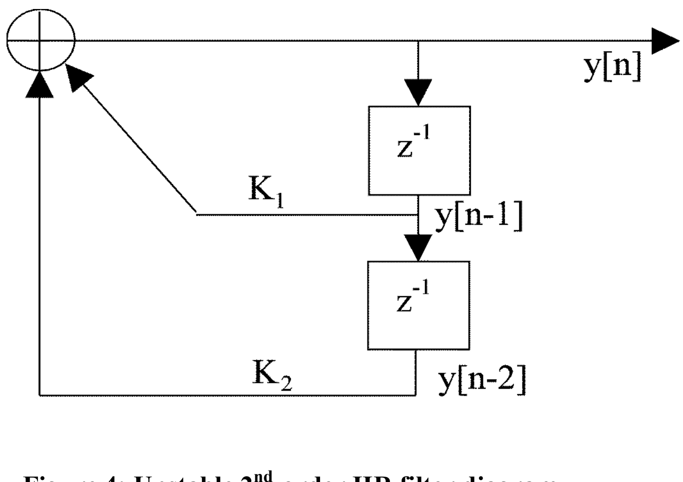


Figure 4: Unstable 2nd-order IIR filter diagram

The equation describing the behavior of the 2nd-order digital filter is:

$$y[n] = K_1 y[n-1] + K_2 y[n-2] + Cx[n]$$

The impulse response ($x[n]=\delta[n-1]$) of the filter is a sinusoidal signal $\sin(2\pi f_c n T_s) = \sin(\omega_c n)$, where T_s is the sampling period and f_c (Hz) and ω (rad) correspond to the analog and digital frequencies, respectively. The coefficients K_1, K_2 and C are determined by:

$$K_1 = 2\cos(\theta);\ \theta = 2\pi\frac{f_c}{f_s} = 2\pi f_c \times T_s$$

$$K_2 = -1$$

$$C = \sin(\theta)$$

These values guarantee that the poles of the filter are located on the unit circle at the digital frequencies ±θ radians.

The value of θ is conditioned by the Nyquist criterion. Hence, the maximum frequency f_c is upper bounded by one half of the sampling frequency ($(f_c)_{max} = f_s/2$). Moreover, it is worth mentioning that when θ decreases, the quality of the generated sinusoidal signal increases because the number of samples per period increases.

Numerical example:

If the sampling frequency is $f_s = 1/T_s = 8000$ Hz, and we wish to generate an f_c =800 Hz sinusoidal signal, then:

$$\theta = 2\pi\frac{f_c}{f_s} = 2\pi f_c \times T_s = \frac{\pi}{5}$$

$$K_1 = 2\cos(\theta) = 2 \times 0.809017;$$

$$K_2 = -1$$

$$C = \sin(\theta) = \sin(\pi/5) = 0.58778$$

With these values, the difference equation can be re-written as:

$$y[n] = 2 \times 0.809017\, y[n-1] - y[n-2],$$

Where the initial value is $C = 0.58778$.

DSP Programming

Main program. To program this sub-system, you need to program using channel 0:

```
Accumulator = K1*Sin-Y2;
Y2 = Sin;
Sin = Accumulator;
```

To try this sub-system, write the result at the output of the DSP. Note that the variable Sin is in floating-point representation, while the output of the DAC must be of type signed integer. Therefore, it is necessary to scale the output of Sin. Since, the value of Sin is limited between -1 and 1, you only need to convert it in Q15 representation.

To observe the generated sinusoidal signal, write the value of Sin multiplied by 2^15-1 (32767) in the variable Output in the line corresponding to the data transmission. You should observe a sinusoidal signal with frequency 5KHz.

Recalculate the values of the relevant parameters to obtain at the output a sinusoidal signal of frequency 10 Khz.

4.4.2 Input acquisition and sign extraction sub-system

The accumulator is loaded with the data at the input of the codec with the instruction DSK6713_AIC23_read(hCodec,&Channel); already given in the program.

- Extract the sign of each data acquired in **Channel** using the logical operation AND (&). To do so, you have to take into account in which part of **Channel** is stored the sign information, and you have to put all the other bits to zero using the AND operation.
- Save the result in the variable **Symbol**.

```
Symbol = ( Channel & 0x8000 ) == 0;
```

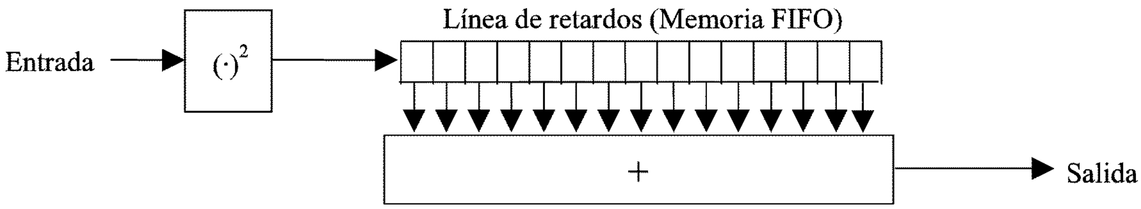
- To verify the functionality of this sub-system, write the content of the **Symbol** multiplied by 2^15-1 at the output of the DSP. Introduce an appropriate signal in the DSP and observe the output.

4.4.3 ASK modulation sub-system

This sub-system reduces to multiplying the carrier signal by the sign of the input data (1 for +1 and 0 for -1).

7.5. Construction of a non-coheren ASK receiver

The second part of this practice consists in the implementation of a non-coherent ASK receiver for the transmitter programmed during the first part of this practice. The structure of this receiver is as follows:



Non-coherent reception of amplitude modulation signals, such as ASK signals, is a very simple operation that consists in processing the input signal using a squaring operator. It is necessary that all the information be contained in the amplitude, since all the information embedded in the phase is lost in this process.

The squaring operator

A squaring operator simply squares the input signal. For an input signal of the form $y(t) = \frac{1}{2}\{1 - \text{signo}(x)\}\cos\omega t$, the output signal of the squaring operator is: $v(t) = \frac{1}{8}(1 - sign(x))^2(1 + \cos 2\pi 2 f_c t) = \frac{1}{8}(1 - sign(x))(1 + \cos 2\pi 2 f_c t)$.

Filtering

The obtained signal contains the desired modulated information in the first term on the right hand. If this signal is filtered to eliminate the frequency $2f_c$, the resulting signal will be 1/8 for positive values of x and zero for negative values of x. To filter this signal, we propose to use an FIR filter with 16 constant coefficients, which simply consists of a First-In First-Out (FIFO) memory. All the contents of the filter buffer are added every time instant. The output of the zero-mean $2f_c$ frequency component will produce an output amplitude smaller or equal to the input signal. On the other hand, the information component (with mean 1/8) will produce an output 16 times larger. This corresponds to multiplying the input signal by 64, determining a 24 dB difference between the $2f_c$ and the information component.

You are expected to implement a program according to the following structure:

```
;       Recibe la señal y la guarda en la memoria
        **** Codificar ****

;       Eleva la señal al cuadrado
        **** Codificar ****

;       Filtra el resultado
                **** Codificar ****
;                       pone el resultado en el primer elemento de la memoria FIR
;                       suma el contenido de todos los elementos de memoria FIR
;                       desplaza todos los elementos de la memoria FIR

;       Emisión de la señal filtrada
;       Programación del CODEC
```

4.5.1 Reception and and squaring operation

- It is necessary to initialize the variable Accummulator. Then, you need to load this register with the input signal.
- To square the acquired signal, you may use a multiplication instruction.
- To verify the correct operation of the program, modifiy the interruption function to observe the squared signal at the output. You must observe a signal of double frequency of the input signal.

4.5.2 Filtering: Version A (FIR with constant coefficients)

- The provided program already implements this type of filtering. Compile the program and run it.
- Check the maximum transmission bit rate of the design system.
- Observe the characteristics of the demodulated signal. At low transmission bit rate (200 bits/s), observe the ripple present in the high value of the signal. Explain this behavior.
-

4.5.3 Filtering: Version B (FIR with variable coefficients)

Although the FIR filter with constant coefficients is attractively very simple, it may be desirable to improve the performance of the FIR filter by using non-rectangular windows. In this section, we will use successively a triangular, a Hanning and a Hamming window.

- Using Matlab, calculate the coefficients corresponding a triangular, Hamming y Hanning windowing.

- Use them to implement an FIR filter. Compare their performance, in terms of the ripple and supported transmission bit rate.
- Characterize the obtained demodulators in terms of maximum and minimum transmission bit rate.
- Using the function FIR1, calculate the coefficients of a low-pass filter with cut-off frequency 3fc/2. Compare the performance of this filter with respect to the previously used windows. Characterize the behavior of this filter.
- For your report, characterize using Matlab the transfer function of the various considered filters and explain their respective behaviors.
- With the filter FIR1, estimate the theoretical maximum transmission bit rate.