TRABAJO PRÁCTICO N° 3
ADC, DAC, FLEXTIMER, DMA, CMP

Se deben implementar dos versiones de un modem FSK, cuya capa física sea similar a Bell 202.
Esta modulación es utilizada en la transmisión de *CallerID* y en la industria (protocolo HART) para conectar sensores a transmisores/PLCs.

1. Descripción

1.1. Detalles de la modulación Bell 202

- Tiempo de bit: 1/1200 s = ~833us
- Frecuencias de *mark* y *space*: 1.2kHz y 2.2kHz respectivamente:

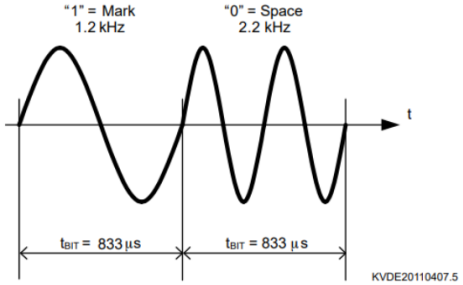


Figure 5. Modulation Timing

Imagen cortesía de la hoja de datos del NCN5193

- Fase continua.
- Ejemplo de señal modulada:

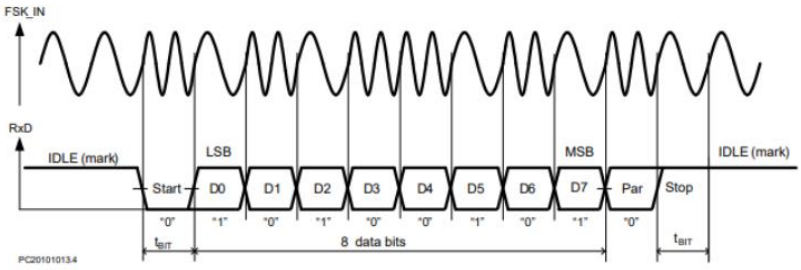
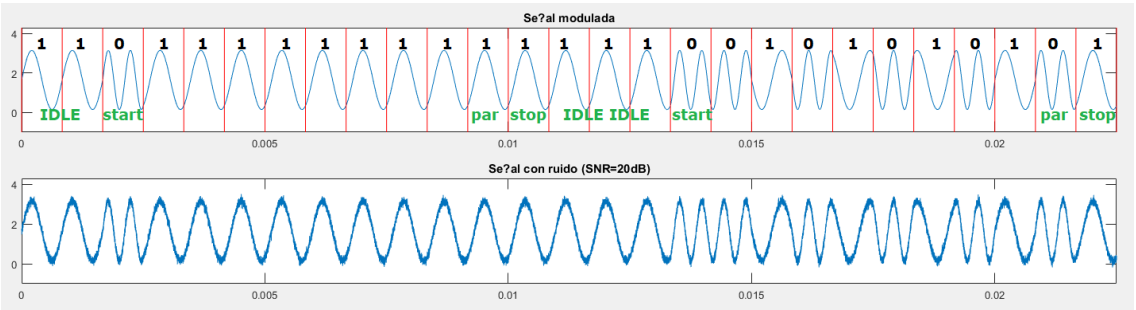


Figure 7. Modulation Timing

This HART bit stream follows a standard 11-bit UART frame with Start, Stop, 8 Data – and 1 Parity bit (odd). The communication speed is 1200 baud.

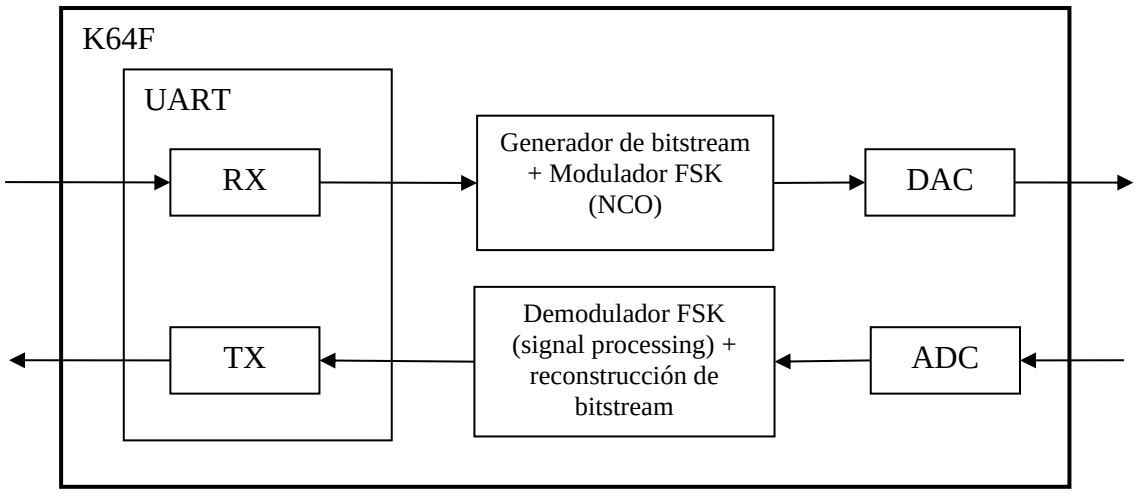
Imagen cortesía de la hoja de datos del NCN5193

- Ejemplo: bits=1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 1 0 1 0 1 0 1



1.2. Versión 1

El diagrama en bloques del equipo es:



Cadena de transmisión:

- El equipo recibirá bytes via la UART, a 1200bps, 8o1 (notar paridad).
- Los caracteres recibidos serán convertidos a un *bitstream* y enviados al modulador.
- El modulador generará la señal modulada FSK y la enviará al DAC, cuyo nivel de DC sea VCC/2 = 1.65V y con amplitud máxima (sin distorsión).

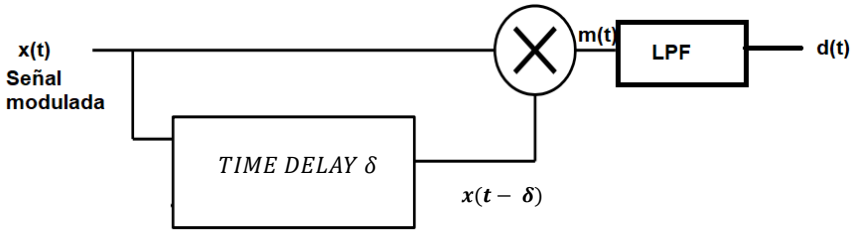
Cadena de recepción:

- El equipo recibirá una señal FSK, la muestreará (ADC) y la enviará al demodulador.
- Luego de la demodulación, se reconstruirá el *bitstream* original, y se generarán los bytes a enviar
- El equipo enviará bytes via la UART, a 1200bps, 8o1 (notar paridad).

1.2.1. Detalles de implementación de la cadena de recepción

Se recibe una señal (posiblemente con ruido) y se muestrea con el ADC. Usar una frecuencia de muestreo adecuada (ej. 12kHz).

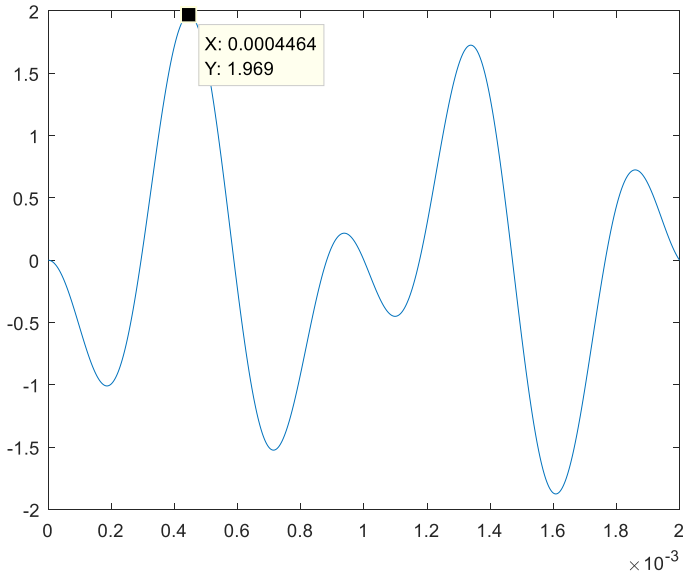
- Se demodula usando el siguiente método:



- La entrada x(t) se multiplica por si misma retardada x(t-δ).
- Si la señal es senoidal, centrada en 0, entonces:
  x(t)=Asin(ωt) y x(t-δ)=Asin(ω(t-δ)).
- El producto m(t)=0.5 A^2 [cos(ωδ) – cos(2ωt + ωδ)].
- Si aplicamos un pasabajos para eliminar el segundo término, nos queda:
  d(t) = 0.5A^2cos(ωδ). Por lo tanto, la salida del filtro pasabajos es función la frecuencia de entrada.

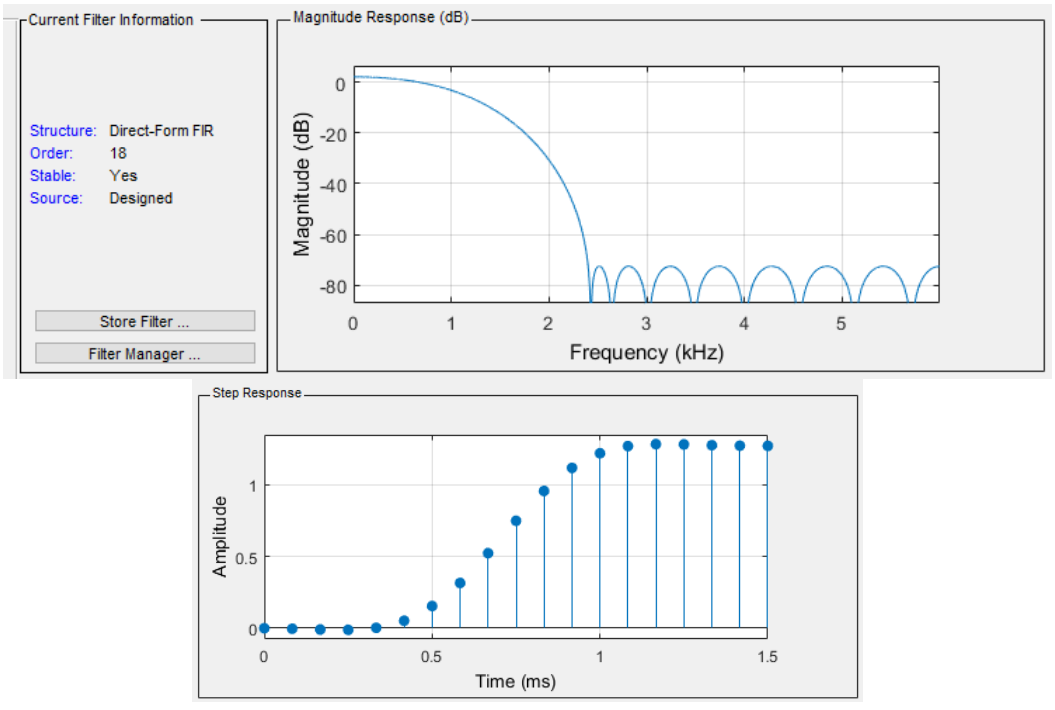
- Para discriminar óptimamente las dos frecuencias f_L=1200Hz y f_H=2200Hz, maximicemos la diferencia entre $d_H(t)$ y $d_L(t)$:

```
δ = linspace(0, 2e-3, 1000);
plot(t, cos(2*pi*2200*δ) - cos(2*pi*1200*δ));
```



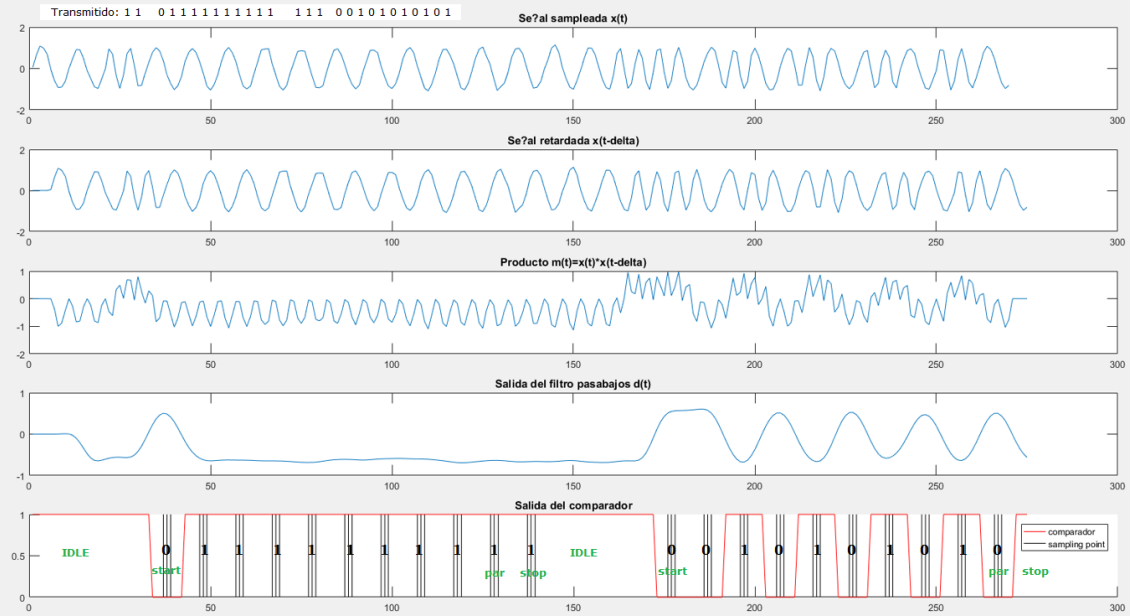
El retardo óptimo (menor a 2ms) es δ=446us. Se podría encontrar un retardo óptimo > 2ms, pero con este valor la diferencia está muy cerca de su óptimo, 2.

- El filtro pasabajos deberá estar dimensionado para atenuar poco la frecuencia correspondiente a la tasa de bits (1200bps), y atenuar suficientemente la frecuencia de 2*f_L=2400Hz. Abajo se muestra la respuesta de un FIR de orden 18, diseñado con *fdatool*.



Tip: cuidar que no haya *overshoot* excesivo en la respuesta al escalón.

- A la salida del filtro se aplica un comparador (posiblemente con histéresis), cuya salida representa el *bitstream*. La cadena de procesamiento es:



- La salida del comparador deberá ser interpretada por un algoritmo similar al que usa un periférico UART para recibir caracteres: *oversampling*. Ver líneas verticales negras en el gráfico "Salida del comparador".

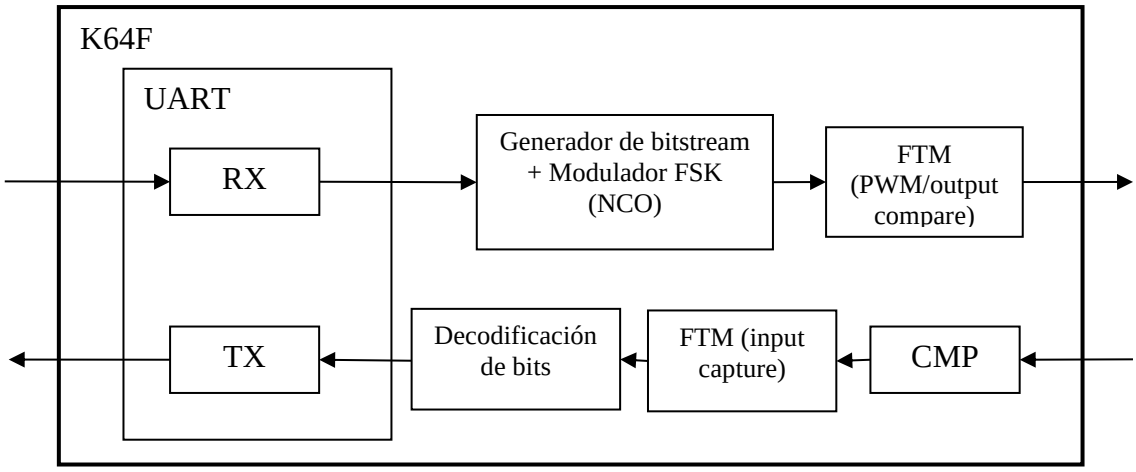
- Una vez identificados los bits, se debe "armar" los caracteres (detectar el *framing*) y enviarlos por UART.

1.3. Versión 2

Hay tres cambios importantes respecto de la versión 1:

1. Se debe reemplazar el demodulador por uno más simple: CMP (comparador **interno**) + FTM (input capture).
2. Se debe **cambiar f_H de 2200Hz a 2400Hz**.
3. Se debe reemplazar la generación de señal (DAC) por una señal generada con timer (FTM: PWM / output compare).

El diagrama en bloques del equipo es:



1.3.1. Detalles de implementación de la cadena de transmisión

- La señal deberá ser generada con el módulo FTM procurando que posea la menor THD posible, sin sobrecargar el uso de la CPU.
- Se debe mantener la propiedad de fase continua.
- La señal que genere el Kinetis debe ser filtrada por un filtro de 1er orden (RC).

1.3.2. Detalles de implementación de la cadena de recepción

- El comparador analógico se usará para detectar cruces por cero de la señal modulada (en realidad, serían cruces por 1.65V). Considerar usar un filtro de 1er orden, RC, a la entrada del comparador para eliminar ruido. Cuidar la respuesta del filtro, para evitar distorsionar excesivamente la señal.
- Se conecta la salida del comparador a un input capture, para medir la frecuencia. Explorar si es posible una conexión interna.
- La frecuencia se interpreta como *mark* o *space*.
- Una vez identificados los bits, se debe "armar" los caracteres (detectar el *framing*) y enviarlos por el puerto serie.

2. Otras consideraciones

2.1. DMA

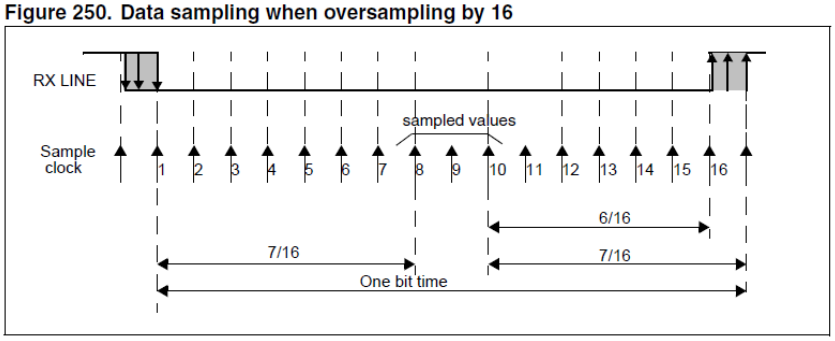
Una y sólo una de las versiones debe ser implementada usando DMA:

- Para la versión 1: DMA para mover datos desde el ADC y hacia el DAC.
- Para la versión 2: DMA para mover datos desde el FTM (input capture) y hacia el FTM (output compare / PWM).

2.2. Tips

- Investigar técnicas de DDS para generar las señales FSK sin saltos de fase.
- Para la versión 1:
  - Se aconseja explorar el uso de la FPU del MCU para hacer eficiente el filtrado FIR.
  - Se aconseja simular la cadena de procesamiento en MATLAB, para entenderla mejor.
  - Estudiar la UART para ver cómo detecta bits 0 y 1 (*oversampling*) e implementar algo similar en software.

Figure 250. Data sampling when oversampling by 16



- Para la versión 2:
  - Investigar el registro SIM_SOPT4.
  - Se sugiere activar la histéresis del comparador (CMP).
  - Se sugiere validar el ancho de pulso capturado para rechazar ruido.

3. Requerimientos

Los requerimientos obligatorios son necesarios para que el trabajo práctico esté aprobado, mientras que los requerimientos opcionales le dan valor (y puntaje) al trabajo, aunque no son necesarios para su aprobación.

3.1. Requerimientos obligatorios

- Correcto funcionamiento de ambas versiones en modo *full-duplex* (las dos placas transmitiendo y recibiendo al mismo tiempo), respetando su frecuencia, nivel DC, amplitud y fase continua. Se evaluará usando dos placas del mismo grupo (no modo *loopback*) transmitiendo y recibiendo un archivo de 1kB.
  Estas pruebas también deberán funcionar con las implementaciones realizadas por los demás grupos.
- Implementación del DMA en una y solo una de sus versiones.
- Informe con mediciones realizadas en un osciloscopio que demuestren el porcentaje de uso de la CPU y máximo tiempo de ISR en ambas versiones.

3.2. Requerimientos opcionales

- Medición del THD de cada una de las versiones. Análisis de cómo generar el PWM de la versión 2 para reducir el THD.
- Para la versión con DMA: incremento del *baudrate* al máximo posible. ¿Hasta dónde llega, "sin" errores? ¿Cuál es la limitación? Justificar dicho valor.
- Para la versión 1: inserción de ruido aleatorio (AWGN) usando el mismo DAC, y medición de la curva BER vs SNR (medir esta curva automáticamente usando un *loopback* con un mismo MCU).
- Implementación de un mecanismo de *Carrier Detect* para mejorar la inmunidad al ruido.

4. Evaluación

Para la nota del trabajo contemplará en orden los siguientes puntos:

1. El funcionamiento del equipo en sus dos versiones.
2. El diseño de los bloques de procesamiento, **implementados de forma eficiente para *streaming***.
3. La medición del **uso de la CPU** y el **máximo tiempo de ISR**, en las dos versiones.