



Las líneas rojas representan el momento donde se debería hacer un sample, corresponde a un sample cada 100µs.  
Las líneas azules indican el momento donde ocurren sucesos representativos entre samples, explicado mas abajo.

Baudrate = 921600  
N = 256  
Fs = 10kHz  
1/Fs = 1/10kHz = 100µs

Samples < N :

1 → 2 : Se reinicia el contador de ciclos. Se muestrea el ADC. (7307 ciclos = 43.49 µs).  
2 → 3 : Se escribe en la UART dos bytes (4723 ciclos = 28.11 µs).  
3 → 4 : Se chequea si ya se enviaron los 256 bytes. Se togglea un led. Se completa el tiempo en un while loop. (4770 ciclos = 28.39µs).  
T<sub>total</sub> = 100µs

Samples == N :

1 → 2 : Se reinicia el contador de ciclos. Se muestrea el ADC. (7307 ciclos = 43.49 µs).  
2 → 3 : Se escribe en la UART dos bytes. Se chequea que Samples = N.(4723 ciclos = 28.11 µs).  
3 → 4 : Se escribe en la UART el header.Se togglea un led. No se completa el tiempo porque ya se paso luego de enviar el header.(36469 ciclos = 217µs).  
T<sub>total</sub> = 288,6µs

Conclusión:

El problema que se tiene es que cuando se llega a **sample == N** se pierden 2 samples debido al tiempo que se tarda en enviar el Header y además no se mantiene la frecuencia de muestreo sino que cada vez que se envía el Header se mueve el sample unos µs antes, es decir, que cada N samples se pierden 2 samples y se adelanta el sample = 0.

Para un baudrate = 460800 el tiempo 3 → 4 es menor debido a que se ocupa mas tiempo para enviar datos por UART en 2 → 3. Alrededor de 16000 ciclos se tarda para sample < N. Para sample == N se tarda 66294 ciclos en enviar el Header por lo tanto se pierden 3 samples.