UNIVERSIDAD DE ALCALÁ ESCUELA POLITÉCNICA SUPERIOR DEPARTAMENTO DE ELECTRÓNICA		GRADO EN INGENIERÍA ELECTRÓNICA DE COMUNICACIONES	
ASIGNATURA	SISTEMAS ELECTRÓNICOS DIGITALES AVANZADOS	FECHA	MARZO 2017
APELLIDOS, NOMBRE		GRUPO	

PRUEBA DE EVALUACIÓN INTERMEDIA 1

Ejercicio 1

El diagrama de bloques de figura 1 muestra un sistema de control a distancia por infrarrojos de servomotor basado en el LPC1768 (Fcpu=100Mhz). El sistema consta de un módulo transmisor y un módulo receptor. El módulo transmisor (TX) dispone de un potenciómetro que permite controlar la posición del servomotor conectado al módulo receptor (RX) en un rango de 0 a 180º variando su recorrido.

El sistema de codificación de infrarrojos se muestra en el Anexo I. Tenga en cuenta que la portadora es una señal periódica cuadrada de 38kHz (zona pintada en negro de la señal). La posición del potenciómetro se codifica con 7 bits y se transmite periódicamente cada 50ms comenzando por el LSB.

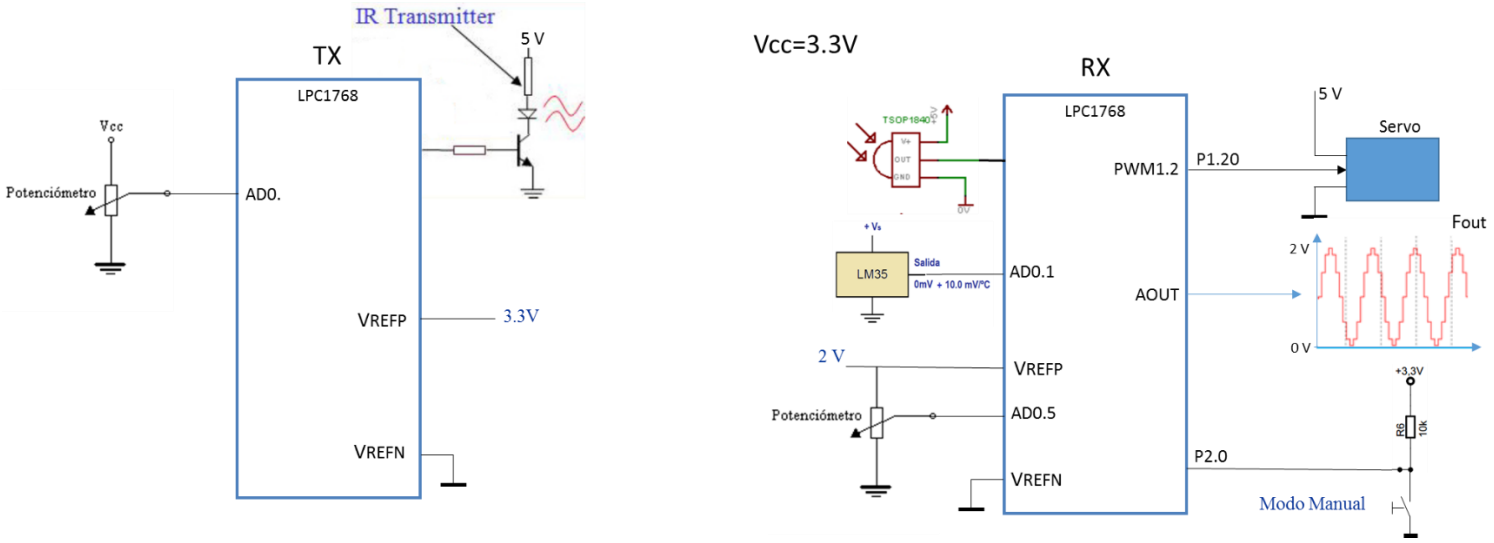


Figura 1. Diagrama de bloques del sistema

El módulo receptor (RX) se encarga de decodificar la señal infrarroja y posicionar el servo en función del código de 7 bits recibido. El potenciómetro conectado a AD0.5 (Modo manual) permite mover el servo en su recorrido (0-180º) mientras se mantiene cerrado el pulsador conectado a P2.0, a la vez que se varía la frecuencia obtenida a la salida del DAC entre 100 y 1 kHz. **En este modo no se tiene en cuenta el código IR recibido.**

Un sensor de temperatura (LM35) adosado al servo nos permite monitorizar su temperatura cada segundo.

a) Complete sobre el diagrama de la figura1, a medida que contesta a los siguientes apartados, el nombre del pin (**Pn.x**) sobre la línea de conexión y el nombre del recurso utilizado dentro del bloque que representa el LPC1768 (ej. **MAT1.0**).

Módulo Transmisor (TX)

b) Complete la función de configuración del ADC (incluyendo los comentarios) para la entrada a la que se conecta el potenciómetro y calcule **tiempo de conversión** para la **mínima frecuencia de reloj** posible del ADC. Escriba en **pseudocódigo** la función de configuración del Timer utilizado para generar una frecuencia de muestreo de **20Hz**. Considere que el ADC funciona por interrupción.

```
void init_ADC_potencimetro(void)
{
LPC_SC->PCONP|= (1<<12);              // Power ON
LPC_PINCON->PINSEL1|= (       );      //
LPC_PINCON->PINMODE1|=(       );      // Deshabilita pullup/pulldown
LPC_ADC->ADCR= (  1<<0)|       //
               (  _<<8)|       //
                (1<<21)|       //    PDN=1
                 (6<<24);      //



}

void init_TIMER ? (void)
{









}
```

c) Complete las líneas de la función de interrupción del ADC para guardar en una variable global el código de 7 bits correpondiente a la posición del potenciómetro.

```
void ADC_IRQHandler(void)
{
   codigo_tx=LPC_ADC->


}
```

d) Explique sin escribir el código la configuración del **Timer 0** que genera exclusivamente la portadora de 38kHz.

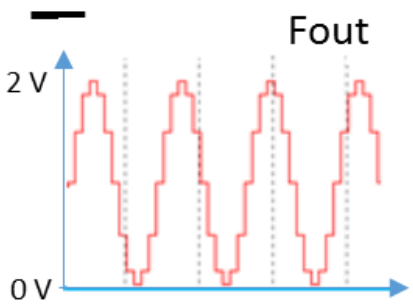
e) Escriba en código o pseudocódigo la función de interrupción del **Timer 2** que permite generar la señal modulada que excita el transistor (ver Anexo I)

Módulo Receptor (RX)

f) Complete la función de inicialización del ADC e indique justificadamente la frecuencia de muestreo de cada canal.

```
void init_ADC(void)
{
LPC_SC->PCONP|= (1<<12);                  // Power ON
LPC_PINCON->PINSEL |= (          );       //
LPC_PINCON->PINSEL |= (          );       //
LPC_PINCON->PINMODE |=(          );       // Deshabilita pullup/pulldown
LPC_PINCON->PINMODE |=(          );       // Deshabilita pullup/pulldown
LPC_ADC->ADCR= (     << )|              //
                  (24_<<8)|              //
                   (1<<21)|              // PDN=1
                    (1<<16);             //
LPC_ADC-ADINTEN=0;                       //
}
```

g) A la vista de señal obtenida a la salida del DAC del módulo receptor, complete la función que genera las muestras de 8 bits de la función seno. Considere declarado el array *uint8_t muestras[ ]*




```
void genera_muestras_seno(void)
{
char i;
for(i=0;i<  ;i++) muestas[i]=(uint8_t)(     +     *sin(2*pi*i/   );
}
```

NOTA: Vrefp= 2V.

h) Escriba la función de interrupción del **Timer 3** que saca las muestras hacia el DAC para generar la señal senoidal.

```
void TIMER3_IRQHandler(void)
{




}
```

i) Explique en detalle, sin escribir código, qué recurso utilizaría y como lo configuraría para reducir la carga de CPU durante la generación de la señal senoidal.

j) Escriba el código de la función de interrupción del SYSTICK (periodo 50ms) que se encarga de leer el estado del pulsador y de modificar la posición del servo en función del potenciómetro, a la vez que modifica el valor de la frecuencia de salida del DAC, así como de almacenar en una variable global la temperatura. Considere que la señal PWM del servo ya está configurada y para un periodo de **15ms**, y que el tiempo a nivel alto varíe entre **0.8-2.4ms**, para un movimiento de su posición entre 0º y 180º.

```
void SysTick_IRQHandler(void)
{




}
```

ANEXO I (Codificación – Decodificación IR)