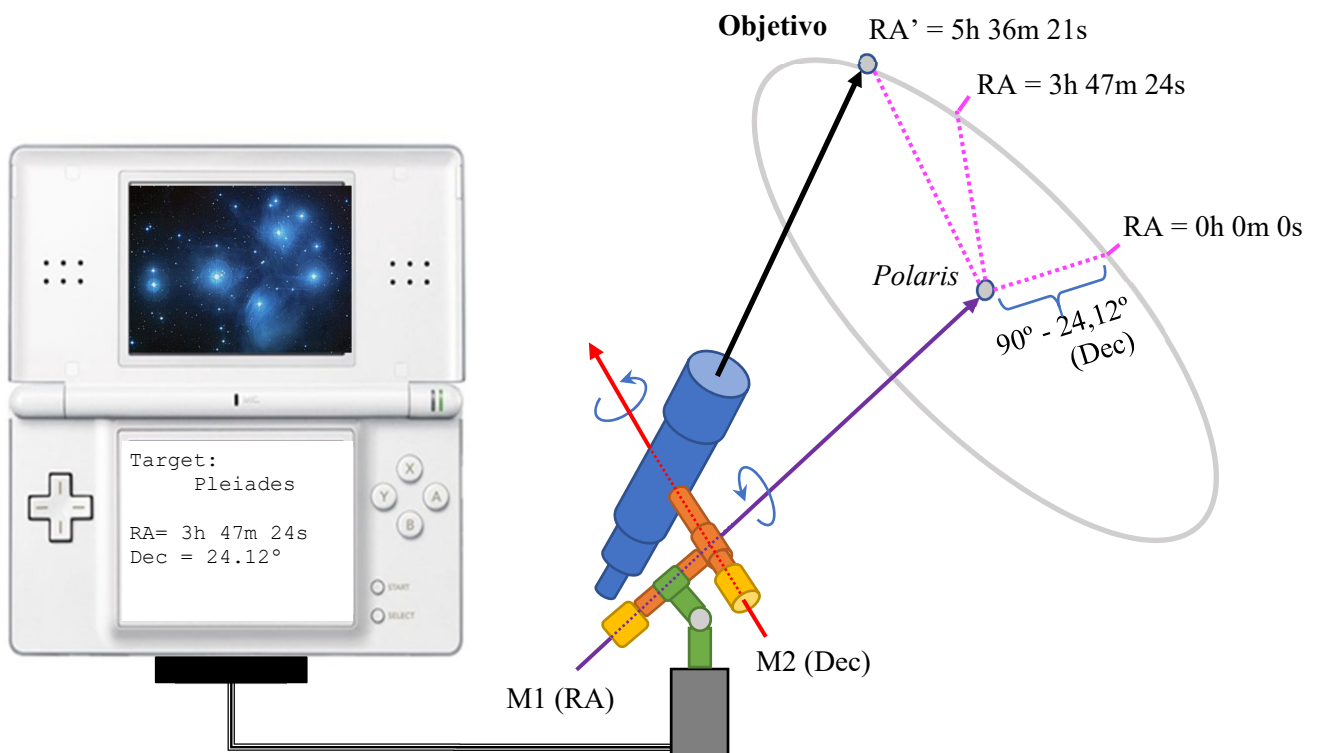


Control de telescopio

Se propone conectar a la NDS un sistema de motores para controlar automáticamente la orientación de un telescopio, utilizando una montura de tipo ecuatorial. Con este tipo de montura, cualquier objeto del firmamento se localiza a partir de dos coordenadas, RA (*Right Ascension*) y Dec (*Declination*). La coordenada Dec se codifica como un ángulo entre -90° y 90° sexagesimales, que indica la elevación respecto al ecuador de la Tierra, de modo que 90° corresponde al Polo Norte y -90° al Polo Sur. La coordenada RA se codifica en horas, minutos y segundos, tomando como inicio la dirección hacia el Sol cuando la Tierra está en el equinoccio de primavera, con un rango total de 24 horas para los 360° de circunferencia.

Las coordenadas RA y Dec son fijas para cada objeto celeste, pero como la Tierra está rotando, la orientación del telescopio hacia esa dirección se debe ir actualizando continuamente para compensar el giro aparente del firmamento. Esta es la principal ventaja de este tipo de montura, que debe estar correctamente orientada hacia el Polo Norte (o Sur) para que se pueda realizar el seguimiento de un objeto variando únicamente la coordenada RA. La coordenada Dec, sin embargo, no se debe modificar una vez localizado el objeto.



En el esquema anterior se muestra la montura correctamente orientada hacia la estrella Polar (*Polaris*), además de los dos motores que controlan la orientación: M1 (RA) y M2 (Dec). El objeto localizado en el ejemplo es el clúster de estrellas Pléyades, en la constelación de Tauro. Sus coordenadas son $RA = 3h\ 47m\ 24s$ y $Dec = 24,12^\circ$, pero en el momento de la observación el telescopio está apuntando a $RA' = 5h\ 36m\ 21s$, porque se supone que ha pasado alrededor de 1 hora y 49 minutos desde que se inició el seguimiento de este objeto. La separación respecto a la orientación Norte se expresa como $90^\circ - 24,12^\circ$ (Dec).

Se dispone de las siguientes rutinas ya implementadas:

<i>Rutina</i>	<i>Descripción</i>
<code>inicializaciones()</code>	Inicializa el <i>hardware</i> (pantalla, interrupciones, etc.)
<code>activar_timer(int ntim, short div_freq)</code>	Activa el funcionamiento e interrupciones del <i>timer</i> <code>ntim</code> , con frecuencia de entrada mínima (32728 Hz) y con el divisor de frecuencia <code>div_freq</code>
<code>tareas_independientes()</code>	Tareas que no dependen de la gestión de los motores del telescopio, por ejemplo, captura y procesamiento de las imágenes obtenidas con una cámara conectada al ocular del telescopio (tiempo de ejecución < 100 ms)
<code>gestionar_interfaz(int *target_ra, short *target_dec)</code>	Gestiona una interfaz de usuario que permite seleccionar un objeto celeste o unas coordenadas concretas, devolviendo un resultado (<i>unsigned char</i>) igual a 0 si no se ha seleccionada nada, igual a 1 si el usuario ha seleccionad un objeto a buscar, registrando sus coordenadas por referencia en <code>target_ra</code> y <code>target_dec</code>
<code>circ_sub(int ra2, int ra1)</code>	Realiza la resta de dos valores de RA (<code>ra2 - ra1</code>) expresados en segundos totales (86.400 segundos = 360°), teniendo en cuenta la circularidad de estos valores, es decir, obtiene el ángulo de separación mínimo entre los dos ángulos correspondientes a <code>ra2</code> y <code>ra1</code> ; devuelve la resta con signo por <code>R0</code> ([<code>-43.200..43.199</code>]), y el valor absoluto de la resta por <code>R1</code> ([<code>0..43.200</code>]), todo expresado en segundos
<code>swiWaitForVBlank()</code>	Espera hasta el próximo retroceso vertical
<code>actualizar_pantallas()</code>	Actualiza el contenido de las pantallas de la NDS, mostrando las variables globales de gestión del telescopio y las imágenes del objeto en observación

El programa a realizar debe permitir entrelazar la ejecución de las tareas independientes con la gestión de la interfaz de usuario y la búsqueda o seguimiento de un determinado objeto celeste.

Para realizar la búsqueda se deberán controlar los motores M1 y M2, mientras que para realizar el seguimiento solo se deberá controlar el motor M1. Dicho control se realizará enviando pulsos por unos determinados bits de un registro de Entrada/Salida a cierta frecuencia (ver descripción más adelante). Para que estos pulsos se puedan enviar concurrentemente con el resto de las tareas, se deberán generar desde las RSIs de los *timers* 1 y 2, respectivamente para cada motor. Además, para realizar el seguimiento se deberá ir reajustando la coordenada RA del objetivo continuamente, con la RSI del *timer* 0.

Se propone el siguiente programa principal:

```
#define FREQ_ENT 32728 // frecuencia de entrada mínima (Hz)

short divfreq_vmax = ¿_a_?; // divisor freq. velocidad máx.
short divfreq_vmin = ¿_b_?; // divisor freq. velocidad mín.

short dec_actual = 0, dec_objetivo = 0; // [-9000..9000] centígrados
int ra_actual = 0, ra_objetivo = 0; // [0..86399] segundos
unsigned char seek_ra = 0, seek_dec = 0; // búsqueda de objetivo
unsigned char track = 0; // seguimiento de objetivo

int main()
{
    inicializaciones();
    activar_timer(0, ¿_c_?); // timer 0 siempre activo
    do
    {
        tareas_independientes();
        if (gestionar_interfaz(&ra_objetivo, &dec_objetivo))
        {
            // activar giros de búsqueda
            activar_timer(1, divfreq_vmax); seek_ra = 1; track = 0;
            activar_timer(2, divfreq_vmax); seek_dec = 1;
        }
        swiWaitForVBlank();
        actualizar_pantallas();
    } while (1);
    return(0);
}
```

Básicamente, el programa utilizará dos variables globales `ra_objetivo` y `dec_objetivo` para determinar la posición del objeto a observar, junto con otras dos variables globales `ra_actual` y `dec_actual` para saber en qué dirección está apuntando el telescopio en cada instante. Las variables DEC almacenan centésimas de grado, mientras que las variables RA almacenan segundos totales, es decir, toda la circunferencia se puede codificar con valores entre 0 y 86.399 segundos ($24 \text{ h} \cdot 60 \text{ m/h} \cdot 60 \text{ s/m} - 1\text{s}$).

En el caso de `ra_actual`, podrá tener valores negativos, por ejemplo, -43.200 segundos equivaldrían a -180° , o positivos iguales o superiores a 86.400, por ejemplo, 129.600 segundos equivaldrían a 540° ($360+180^\circ$). Esto puede pasar cuando se incremente o decremente esta variable para aproximarse a la posición objetivo (`ra_objetivo`). Esto no supondrá ningún problema a la hora de comparar dichas variables, si se utiliza la rutina `circ_sub()`, que tiene en cuenta la propiedad de circularidad de estas variables.

Además, otras tres variables globales booleanas determinan si se está moviendo el telescopio para fijar las coordenadas RA y DEC (`seek_ra` y `seek_dec`), o si se está moviendo el telescopio para compensar la rotación de la Tierra (`track`).

Como se puede observar, se han dejado unas marcas `¿_a_?`, `¿_b_?` y `¿_c_?` para indicar diversos divisores de frecuencia, que se piden como parte de la solución.

Por otro lado, se supone que las RSIs de los *timers* 0 y 2 ya están implementadas, con lo cual, solo hay que implementar la RSI del *timer* 1.

La RSI del *timer* 1 debe controlar los bits 5, 6 y 7 del registro REG_TEL, que tienen las siguientes funcionalidades:

Control	M2:	M1:	
	bit 1	bit 5	0 -> giro grueso, 1 -> giro fino
	bit 2	bit 6	0 -> incrementar, 1 -> decrementar
	bit 3	bit 7	pulso -> una variación de giro

Los motores pueden avanzar en dos resoluciones, según la selección del tipo de giro:

giro grueso:	1 pulso = 0,5°	(120 segundos de RA)
giro fino:	1 pulso = 0,00417°	(1 segundo de RA, aprox.)

La frecuencia máxima de pulsos que aceptan los motores, independientemente del tipo de giro, es de 15 pulsos por segundo. Cada pulso debe ser simétrico, es decir, el mismo tiempo a 1 que a 0. A la velocidad (frecuencia) máxima y con giro grueso, el motor M1 puede dar una vuelta completa en 48 segundos. Además, el motor M1 puede compensar la rotación de la Tierra si se configura en giro fino, con un pulso de incremento por segundo.

En general, la RSI del *timer* 1 se debe encargar de las siguientes tareas:

- generar pulsos por el bit 7, a la frecuencia adecuada,
- fijar el sentido de giro (incremento o decremento), según los valores de `ra_objetivo` y `ra_actual`,
- en modo de búsqueda, si la diferencia entre `ra_objetivo` y `ra_actual` es superior o igual a 120 segundos, usar el giro grueso a la velocidad máxima,
- en modo de búsqueda, si la diferencia entre `ra_objetivo` y `ra_actual` es inferior a 120 segundos y superior a 0 segundos, usar el giro fino a la velocidad máxima,
- en modo de búsqueda, si `ra_objetivo` = `ra_actual`, pasar a modo seguimiento, reconfigurando el *timer* 1 para girar el motor M1 a la velocidad mínima (1 pulso/seg),
- en modo de seguimiento, usar el giro fino a la velocidad mínima,
- actualizar el valor de `ra_actual`, teniendo en cuenta el tipo de giro (grueso o fino) y el sentido de giro (incremento o decremento).

Por su parte, la RSI del *timer* 0 irá incrementando el valor de `ra_objetivo` continuamente, teniendo en cuenta la propiedad de circularidad (cuando llegue a 86.400 pasará a 0).

Se pide:

RSI del *timer* 1 en ensamblador, expresión del cálculo de los divisores de frecuencia `_a_`, `_b_` y `_c_` en lenguaje C.