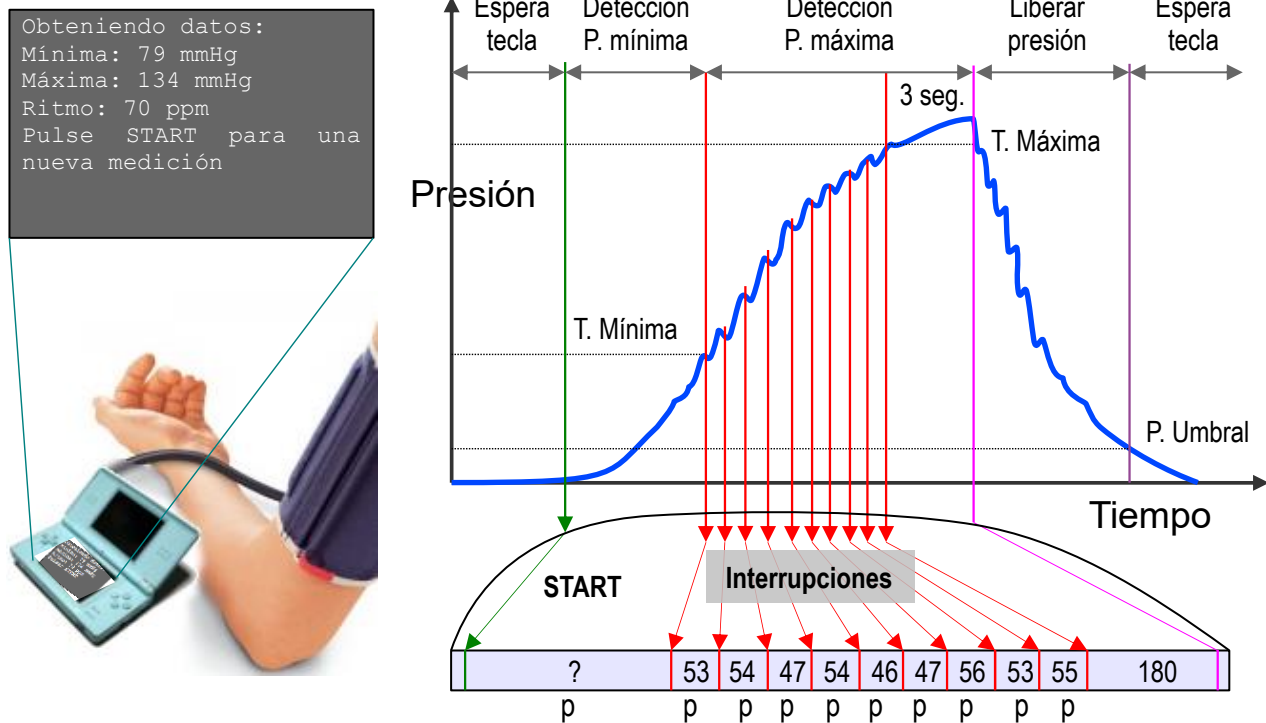


## Problema 13: Tensiómetro (Ex. 2ª Conv. 2013-14)

Se propone controlar con la NDS un dispositivo para medir la tensión arterial de una persona:



El funcionamiento del sistema será el siguiente:

1. Inicializar el sistema.
2. Presentar mensaje de inicio: "Pulse START para iniciar la medición".
3. Detectar pulsación de la tecla 'START'.
4. Borrar pantalla y presentar mensaje: "Obteniendo datos:".
5. Iniciar el ciclo de aumento de presión: cerrar la válvula del dispositivo y activar un motor que insufla aire dentro del brazalete neumático.
6. Cuando haya suficiente presión, el dispositivo empezará a detectar latidos del corazón; la tensión mínima será la presión del brazalete al detectar el primer latido.
7. Se continuará aumentando la presión hasta que no se detecten más latidos; además, durante esta fase del proceso se debe calcular el tiempo entre cada dos latidos, almacenándolo en un vector de periodos ( $p_0, p_1, p_2, \dots, p_{n-1}, n < 50$ ) para después poder calcular el ritmo cardíaco.
8. Cuando el tiempo desde que se detectó el último latido sea superior a 3 segundos se considerará que se ha superado la presión máxima; la tensión máxima será la presión del brazalete al detectar el último latido.
9. Presentar por pantalla los resultados: tensión mínima, tensión máxima y ritmo cardíaco.

10. Iniciar ciclo de disminución de presión: abrir la válvula del dispositivo, parar el motor y esperar a que la presión del brazalete llegue a un valor umbral mínimo (10 mmHg).
11. Presentar mensaje de continuación: “Pulse START para una nueva medición” y repetir todo el proceso desde el paso 3.

Este dispositivo periférico presenta dos registros de Entrada/Salida de 16 bits:

- TENS\_CTRL: 1 → cierra válvula y activa motor para insuflar aire, 0 → abre válvula para liberar el aire y para motor.
- TENS\_DATA: valor actual de la presión del brazalete, expresada en mmHg (milímetros de mercurio).

En el ciclo de aumentar la presión, cada vez que el dispositivo detecta un latido genera una interrupción específica (IRQ\_CART).

Se dispone de las siguientes rutinas, ya implementadas:

<i>Rutina</i>	<i>Descripción</i>
<code>inicializaciones()</code>	Inicializa pantalla e interrupciones
<code>scanKeys()</code>	Captura el estado actual de los botones de la NDS
<code>int keysDown()</code>	Devuelve un patrón de bits con los botones activos
<code>swiWaitForVBlank()</code>	Espera hasta el próximo retroceso vertical
<code>clear()</code>	Borra todo el contenido de la pantalla
<code>printf(char *format, ...)</code>	Escribe por pantalla un mensaje con un formato específico

Para obtener el tiempo entre los latidos del corazón hay que utilizar la rutina `swiWaitForVBlank()` y una variable global de tiempo que se incremente a cada retroceso vertical. Esto proporcionará un valor de período entre dos latidos, si en cada interrupción del tensiómetro se vuelve a poner a cero el tiempo actual. Como puede haber pequeñas variaciones de tiempo entre los latidos detectados, se pide que se almacenen dichos periodos en un vector para posteriormente obtener un valor promedio de todos los periodos.

Dicho valor promedio estará expresado en retrocesos verticales. Por lo tanto, hay que realizar las conversiones oportunas para obtener el ritmo cardíaco del usuario en pulsaciones por minuto (frecuencia).

Suponiendo que dicho ritmo nunca será inferior a 20 ppm ni superior a 200 ppm, los valores límite del periodo promedio serán 180 y 18 retrocesos verticales, respectivamente.

Todos los cálculos del período promedio y del ritmo cardíaco se encapsularán en la siguiente rutina:

```
unsigned char calcular_ritmo(unsigned char periodos[],  
                           unsigned char n_elem);
```

donde el primer parámetro corresponderá al vector de todos los períodos obtenidos (paso por referencia) y el segundo parámetro será el número de elementos registrados en el vector. El resultado que devolverá la rutina será el ritmo cardíaco expresado en pulsaciones por minuto. Para realizar divisiones desde lenguaje ensamblador hay que usar la rutina BIOS `swi 9` (entrada: R0 = numerador, R1 = divisor; salida: R0 = cociente, R1 = resto, R3 = cociente absoluto).

### Se pide:

Programa principal en C, RSI y rutina `calcular_ritmo()` en ensamblador.