

## Práctica 5

### • Manejo de la sección de E/S del microcontrolador ATmega1280/2560

**Objetivo:** Mediante esta práctica el alumno analizará la implementación de retardos por software, así como también se familiarizará con la configuración y uso de puertos.

**Equipo:** - Computadora Personal con AVR Studio y tarjeta T-Juino.

**Teoría:** 1) Análisis y cálculo del retardo por SW de la práctica.  
2) Teoría sobre puertos de E/S (uC ATmega1280/2560)

### Descripción:

Implementar un programa en base al Listado dado en *Prac5\_IO.c*, el cual es un simple juego basado en el tiempo de reacción del jugador. Tiene como entrada un botón y dependiendo de su duración que esta presionado determina la acción en el juego. A su vez se muestra en el arreglo de Leds un patrón que refleja el estado actual del juego; los cuales se deberán conectar como se muestra en la Fig. 2.

### Funciones a implementar:

1. `uint8_t myRand(uint8_t seed)`  
Función que deberá estar definida en ensamblador en un archivo .S, y tendrá que ser llamada desde C.
2. `void delay(uint16_t mseg);`  
Función que debe de tardarse **n ms** en retornar, según se especifique en el parámetro de entrada. Con una exactitud de  $\pm 5 \mu s$ .
3. `void InitPorts(void);`  
Inicialización requerida de los puertos utilizados en esta práctica, según la Fig. 2.
4. `uint8_t check_Btn(void);`  
Retorna el estado del botón, detectando entre NOT\_PRESSED, SHORT\_PRESSED y LONG\_PRESSED. Donde el umbral para una larga duración es cualquiera que sea mayor a 1 s. Ignorando el rebote mecánico que se puede apreciar en la Fig 1.

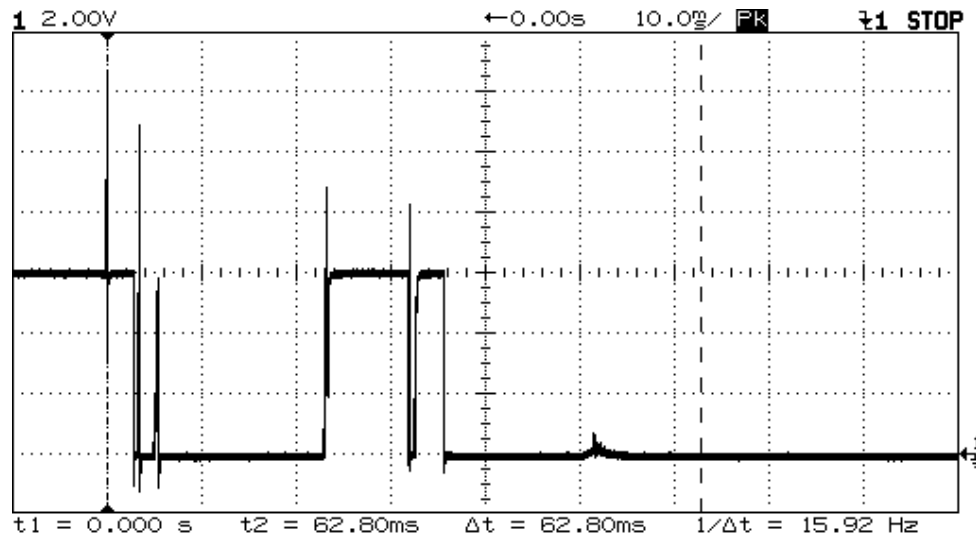


Fig. 1. Ejemplo del rebote mecánico de un botón.

##### 5. void updateLeds(uint8\_t gameState)

Muestra el patrón actual que refleja el estado del juego. Estos estados son los siguientes:

- **eWaitForStart:**  
Secuencia de *walking-cero* del MSB al LSB, actualizándose cada 100 ms  
(s<sub>0</sub>=0b11111111, s<sub>1</sub>=0b01111111, ..., s<sub>8</sub>=0b11111110, y repetir)
- **eStartCount:**  
Secuencia aleatoria que se actualiza cada 100 ms.
- **eEndCount:**  
Todos los LEDs apagados.
- **eYouLoose:**  
Secuencia donde medio *nibble* esta apagado y el otro prendido, alternando cada 500 ms.
- **eYouWin:**  
Secuencia que alterna todos los LEDs se encienden y apagan simultáneamente alternando cada 250 ms.

Realizar los ajustes necesarios de tal forma que no se perciba que parpadean.

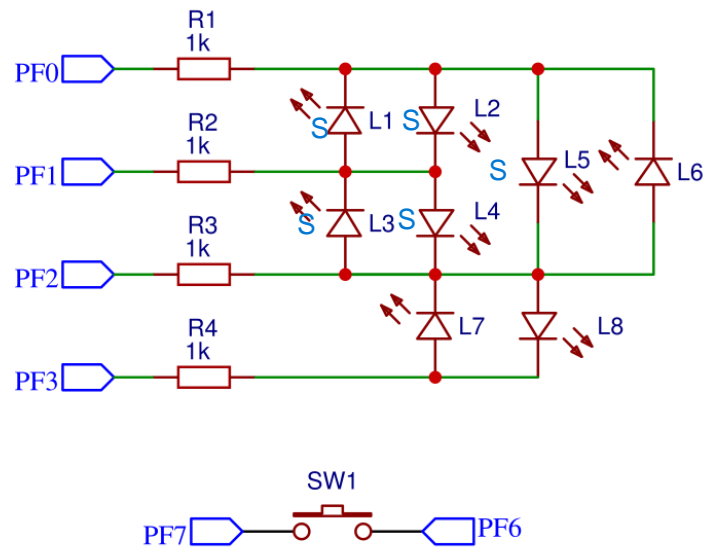


Fig. 2. Esquemático

**Comentarios y Conclusiones.**

**Bibliografía y Referencias.**