

# Circuitos Digitales y Microcontroladores

## Trabajo Práctico N°2

Ramiro Cabral 03226/6

Manuel Savenia 03302/1

### 1. Máquina de Estados

Para manejar los estados del juego, desarrollamos una maquina de estados. La misma es una máquina de Moore, debido a que la salida del sistema depende unicamente del estado actual de la misma. Tenemos 5 estados:

- INICIO
- INICIAR\_JUEGO
- JUEGO
- DERROTA
- VICTORIA

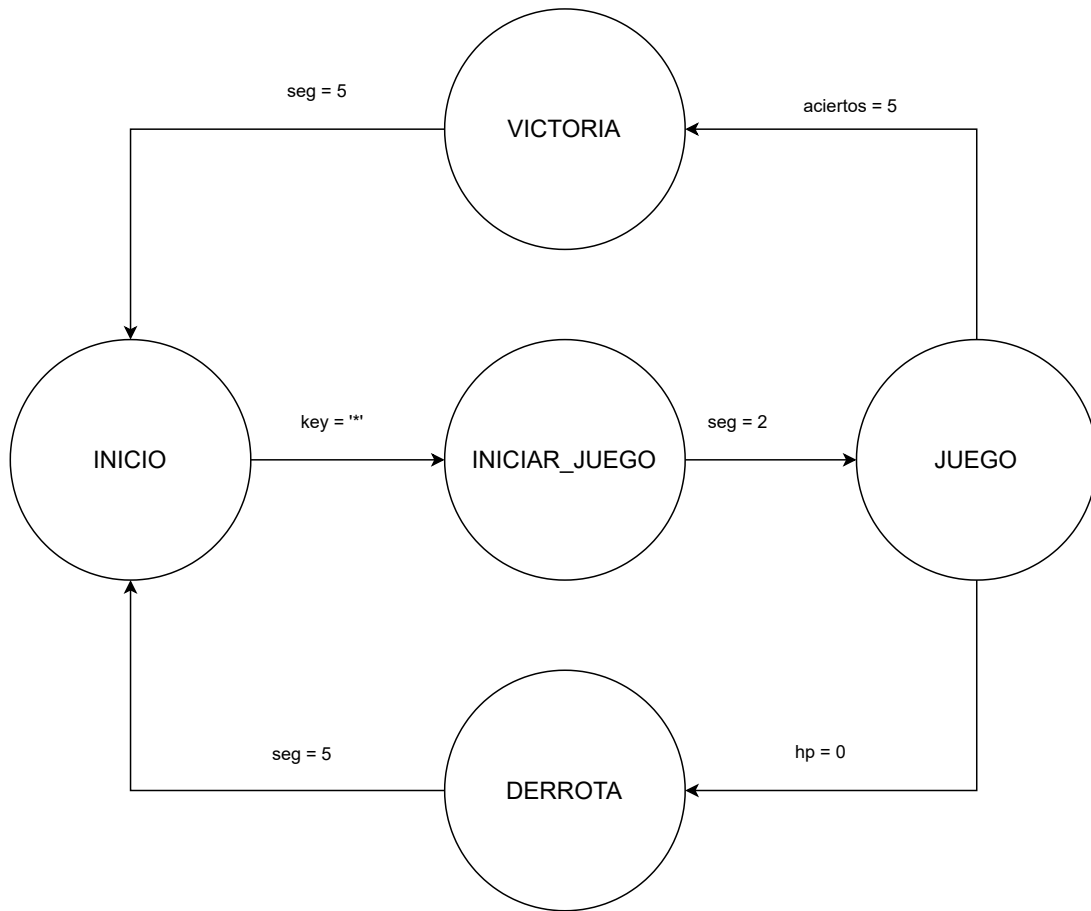


Figura 1: Maquina de Estados.

## 2. Demostracion de tiempos

### Interrupcion de Timer.

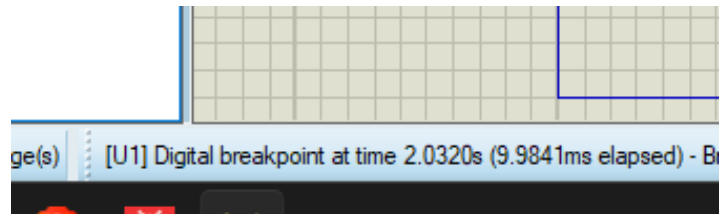
Primero seteamos un breakpoint en la interrupcion del timer:

```

----- #include <avr/io.h>
----- #include <avr/interrupt.h>
----- #include "timer.h"
----- #include "mef.h"
-----
----- void timer0_init()
----- {
-----     // Configurar Timer0 para interrup
00E6     TCCR0A = (1 << WGM01);
00EA     TCCR0B = (1 << CS02) | (1 << CS00)
00EE     OCR0A = 155;
00F2     TIMSK0 |= (1 << OCIE0A);
00FC     sei();
----- }
-----
----- // interrupcion de Timer0, se ejecuta
----- ISR(TIMERO_COMPA_vect)
0100 {
010C     tick++;
0116     flag = 1;
011C     if (tick == 100)
-----     {
0124         tick = 0;
0128         seg++;
-----     }
0132 }

```

Luego, corroboramos que se ejecuta cada 10 *ms*:



Vemos que la ejecucion se ejecuta cada 9,98 *ms*, lo que es mas que aceptable.

## Tiempo de inicio.

Cuando la palabra aparece en la pantalla, se deben esperar 2 segundos hasta que comience el juego.

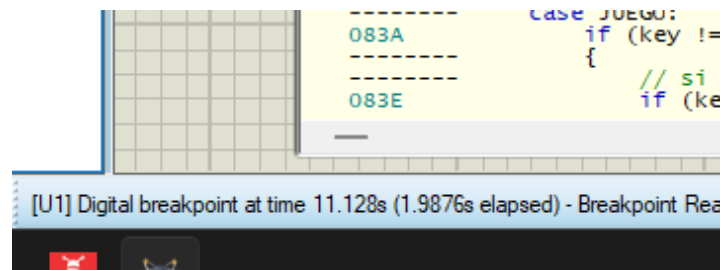
Breakpoint:

```

-----
-----      case INICIAR_JUEGO:
-----      // mostrar password por dos segundos
0780      if (first)
-----      {
-----          LCDclr();
-----          _delay_ms(160);
07CE          LCDGotoXY(0, 0);
07D6          LCDstring(password, 5);
07E4          first = 0;
▶ 07E8          reset_time();
-----      }
-----      // comenzar juego
07EC      if (seg == 2)
-----      {
07F6          curr_state = JUEGO;
07FA          LCDclr();
07FE          pass_index = 0;
0802          input_char = 0;
0806          strcpy(life, "<3<3<3");
0818          reset_time();
081C          LCDGotoXY(0, 0);
0824          LCDstring((uint8_t *)guess, 5); // Mostrar la palabra oculta
082E          first = 1;
-----      }
-----      break;
-----

```

Luego, corroboramos que se muestra por 2 s:



## Tiempo de derrota.

En caso de que se agoten las 3 vidas, estamos ante una derrota. Se debe mostrar la pantalla de derrota junto a la palabra por 5 segundos.

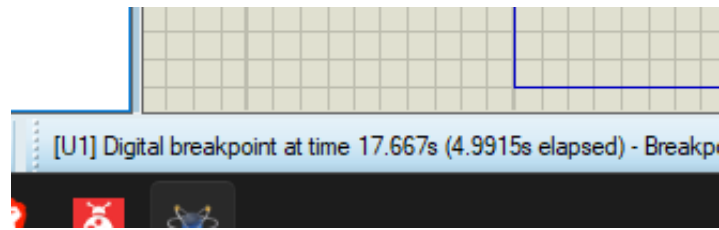
Breakpoint:

```

-----
-----      case DERROTA:
08FE      if (first)
071E      {
-----          LCDclr();
-----          _delay_ms(160);
091C          LCDstring("Derrota!", 8);
0926          LCDGotoXY(0, 1);
092E          LCDstring(password, 5);
093C          first = 0;
0940          reset_time();
-----      }
-----      else
-----      {
-----          // esperamos 5 segundos para volver al inicio
0946          if (seg == 5)
-----          {
▶ 0950              curr_state = INICIO;
0954              reset_time();
-----          }
-----      }
-----      break;
-----

```

Luego, corroboramos que se muestra por 5s:



## Tiempo de victoria.

Si se adivina la palabra, se debe mostrar una pantalla de victoria por 5 segundos.  
Breakpoint:

```

-----      case VICTORIA:
095A          if (first)
-----          {
-----              LCDclr();
-----              _delay_ms(160);
0978              LCDstring("Victoria!", 9);
0982              LCDGotoXY(0, 1);
-----              char message[20];
098A              sprintf(message, "Tiempo: %ds", sec_to_win);
09A8              LCDstring(message, 11);
0980              first = 0;
● 0984              reset_time();
-----          }
-----          else
-----          {
-----              // esperamos 5 segundos para volver al inicio
09C6              if (seg == 5)
-----              {
● 09CE                  curr_state = INICIO;
09D2                  reset_time();
-----              }
-----          }
-----          break;
-----      }
09D6  }

```

Luego, corroboramos que se muestra por 5s:

