Software en Tiempo Real

Msc. Ing. Carlos Centeno
Ingeniería Electrónica
UTN FRC

Año 2023

Condiciones de la Materia

- REGULARIDAD
  - Asistencia del 75% sumadas clases teóricas y clases prácticas.
  - Presentación de Trabajos Prácticos desarrollados sobre un sistema embebido a elección.
- APROBACION DIRECTA
  - Ser Regular.
  - Aprobar Parcial o Recuperatorio con nota Mayor o igual a 7.
- FINAL
  - Presentación de Trabajo Integrador en turno de exámen.

Ciclo de Vida

- Fases de DISEÑO
  - Especificación del producto
  - Definición del Hardware mínimo
    - Evaluación de dispositivos conocidos disponibles
  - Diseño del software acorde al hardware seleccionado.
    - Uso de compilador específico a la plataforma.
      - Conocer las consideraciones particulares de la herramienta elegida.
    - Empleo de emulador y/o simulador.
  - Implementación del Hardware y sus circuitos asociados.
    - Desarrollo y fabricación de prototipo.

Ciclo de Vida

- Fases de DISEÑO
  - Integración del SOFT en el HARD
  - Testeo del sistema embebido en Laboratorio.
  - Generación de Manuales.
  - Test de campo.
  - Generación de Preserie.
  - Generación de Serie.
  - Implementación y/o distribución de Actualizaciones.

Ciclo de Vida

- Especificación del Producto/Sistema
  - Se puede partir desde la definición del sistema hacia sus componentes específicos.
  - Se pueden definir las partes específicas para luego realizar la correspondiente integración.
  - Debe ser precisa y aceptada por el cliente.
  - Deben participar la mayor cantidad de actores involucrados en la posterior implementación.
  - Se deben determinar
    - Requerimientos → Punto de Vista Usuario
    - Especificaciones → Punto de Vista Desarrollador.

Ciclo de Vida

- Definición del HARDWARE
  - Siempre se deben tener en cuenta al momento de iniciar los puntos a futuro que desea potenciar el cliente.
    - Posibilidad de actualización del SOFT.
    - Incremento de capacidades del HARD.
  - Compatibilidad con otras familias.
  - El lenguaje y sus particularidades según la plataforma empleada.
  - Herramientas de desarrollo disponibles.
  - RTOS disponible.

Ciclo de Vida

- Definición del HARDWARE
  - La definición de la plataforma a utilizar nos indica la forma en que se emplearán los recursos disponibles.
  - La definición de la plataforma indicará que periféricos y/o interfaces integradas estarán disponibles.
  - Para el empleo de un recurso es necesaria la programación del mismo, asignado a sus registros particulares de configuración valores determinados. (Uso de RS232 → Velocidad : 9600 bps → Trama: 8N1)

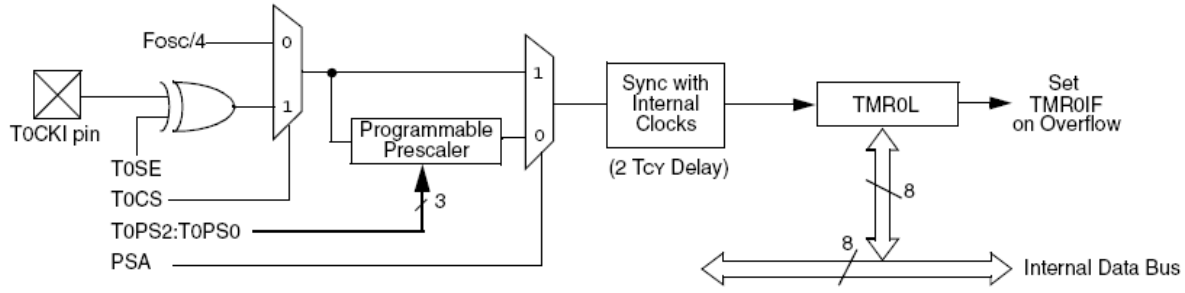
Ciclo de Vida

- Definición del HARDWARE

TABLE 18-3: BAUD RATES FOR ASYNCHRONOUS MODES

BAUD RATE (K)	SYNC = 0, BRGH = 0, BRG16 = 0											
	FOSC = 40.000 MHz			FOSC = 20.000 MHz			FOSC = 10.000 MHz			FOSC = 8.000 MHz		
	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)	Actual Rate (K)	% Error	SPBRG value (decimal)
0.3	—	—	—	—	—	—	—	—	—	—	—	—
1.2	—	—	—	1.221	1.73	255	1.202	0.16	129	1201	-0.16	103
2.4	2.4											
9.6	9.6											
19.2	19.5											
57.6	56.8											
115.2	125.											

FIGURE 11-1: TIMER0 BLOCK DIAGRAM (8-BIT MODE)



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI max. prescale.

Tipos de Variables

- Con Signo
- Sin Signo - UNSIGNED
- CHAR (1byte)
  - 0 a 255
  - -128 a 127
- INT (2 byte)
  - 0 a 65535
  - -32768 a 32767
- FLOAT (4bytes)
  - 1x10 exp 37
  - 1x10 exp -37
- DOUBLE (8bytes)
  - 1x10 exp 308
  - 1 x10 exp -308
- LONG (4 byte)
- SHORT (2 byte)
  - 32767
  - -32768

- Vectores
  - int vector [dimension]
- Matrices
  - int nombre [fila] [columna]
- Estructuras
  - Conjunto de distintos datos

```
struct {
  int dato1;
  char dato2;
  }nombre;
```

  - Forma de uso

```
nombre.dato1 = 10;
```

Estructuras de Control

- FOR

```
for(inicio; condicion; incremento)
{
    operaciones
}
```

- WHILE

```
while(condicion)
{
    operaciones
}
```

- SWITCH

```
switch(variable)
{
    case valor1:
                break;
    case valor2:
                break;
    Default:
}
```

- IF-ELSE

```
if(condición)
    operación
else
    operación
```

Funciones

- TIPOS
  - Sin recepción de parámetros
    void funcion(void)
  - Con recepción de parámetros
    void funcion(int a, char b)
  - Sin devolución de parámetros
    void funcion(void)
    void funcion(int a, char b)
  - Con devolución de parámetros
    int funcion(int a, char b)

```
void delay(int tiempo)
{
    int i=0;
    for(i=0; i <= tiempo * 123; i++)
    {
      #asm
      nop
      nop
      nop
      #endasm
    }
}

int suma(int A, int B)
{
    int valor;
    valor = A + B;
    return valor;
}
```

Punteros a Función

```
int (*funcion)(int, int);
int (*funcion1)(int, int);
int suma (int, int);
int resta(int, int);

void main(void)
{
  funcion = suma;
  funcion1 = resta;
  printf("La suma es %d   y la resta es %d", funcion(3,5), funcion1(32,12));
}

int suma(int a, int b){
  return a+b;
}

int resta(int a, int b){
  return a-b;
}
```

Operaciones a Nivel de BITS

- AND

A	B	Salida
0	0	0
0	1	0
1	0	0
1	1	1

- OR

A	B	Salida
0	0	0
0	1	1
1	0	1
1	1	1

1	0	0	**1**	1	0	0	1
&							
0	0	0	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0

1	0	0	**1**	1	0	0	1
&							
1	1	1	**0**	1	1	1	1
1	0	0	**0**	1	0	0	1

1	0	0	1	**1**	0	1	0
|							
0	0	0	0	**1**	0	0	0
1	0	0	1	**1**	0	1	0

Directivas del Preprocesador

- Son sentencias del compilador que permiten entre otras cosas:
  - Una mayor facilidad al momento de desarrollar código y/o programas
  - Poder leer con facilidad un código
  - Poder modificar con facilidad
  - Una mayor transparencia del código entre diversas arquitecturas de máquinas.

Directivas del Preprocesador

#include “stdio.h”

- Incluye un archivo fuente, en este caso la librería stdio.h para luego poder compilar ambos.

#define MAXIMO 125

- Se usa para definir constantes o macros. Este “rótulo” será reemplazado cada vez que se mencione en el código.

```
#define FALSO 0
#define VERDADERO !FALSO
#define MIN(a,b) (a < b) ? a : b
```

Directivas del Preprocesador

```
#define MEX 0
#define EUA 1
#define FRAN 2

#define PAIS_ACTIVO MEX

#if PAIS_ACTIVO == MEX
    char moneda[]="pesos";
#elif PAIS_ACTIVO == EUA
    char moneda[]="dolar";
#else
    char moneda[]="franco";
#endif
```

Consideraciones Especiales

- Cuando usar RTOS
  - Es conveniente
  - Que recursos necesito
  - Cual es la definición concreta de RTOS
- Cuales son las opciones en el mercado

Super Bucle

- Resolvamos un ejemplo
  - Implementar un Sistema Embebido que controle secuencias temporales en salidas digitales.
  - Usar topología Super Loop.
  - El control de tiempo se realiza con espera pasiva.

Secuencia

- Salida 1:
  - Alto : 1mS
  - Bajo : 1mS
- Salida 2:
  - Alto : 2mS
  - Bajo : 2mS
- Salida 3:
  - Alto : 3mS
  - BAjo: 4mS