**JOHN PAUL NEGRETE HERNANDEZ**

**MORAN GARABITO**

**EMBEBIDOS**

**MECATRONICA 8-B**

**(interruptor)**

****

Introducción:

Las interrupciones son un recurso esencial de los sistemas embebidos. Básicamente, la interrupción es un mecanismo mediante el cual el CPU puede, ante cierto evento, suspender lo que está haciendo en ese momento y pasar a atender una rutina de alta prioridad

Para demostrar el uso de interrupciones en este tipo de microcontroladores se presentan a continuación dos ejemplos. El primero de ellos simplemente repite el programa hola mundo. La configuración del timer se hace de tal manera que se produzcan interrupciones cada 128us.

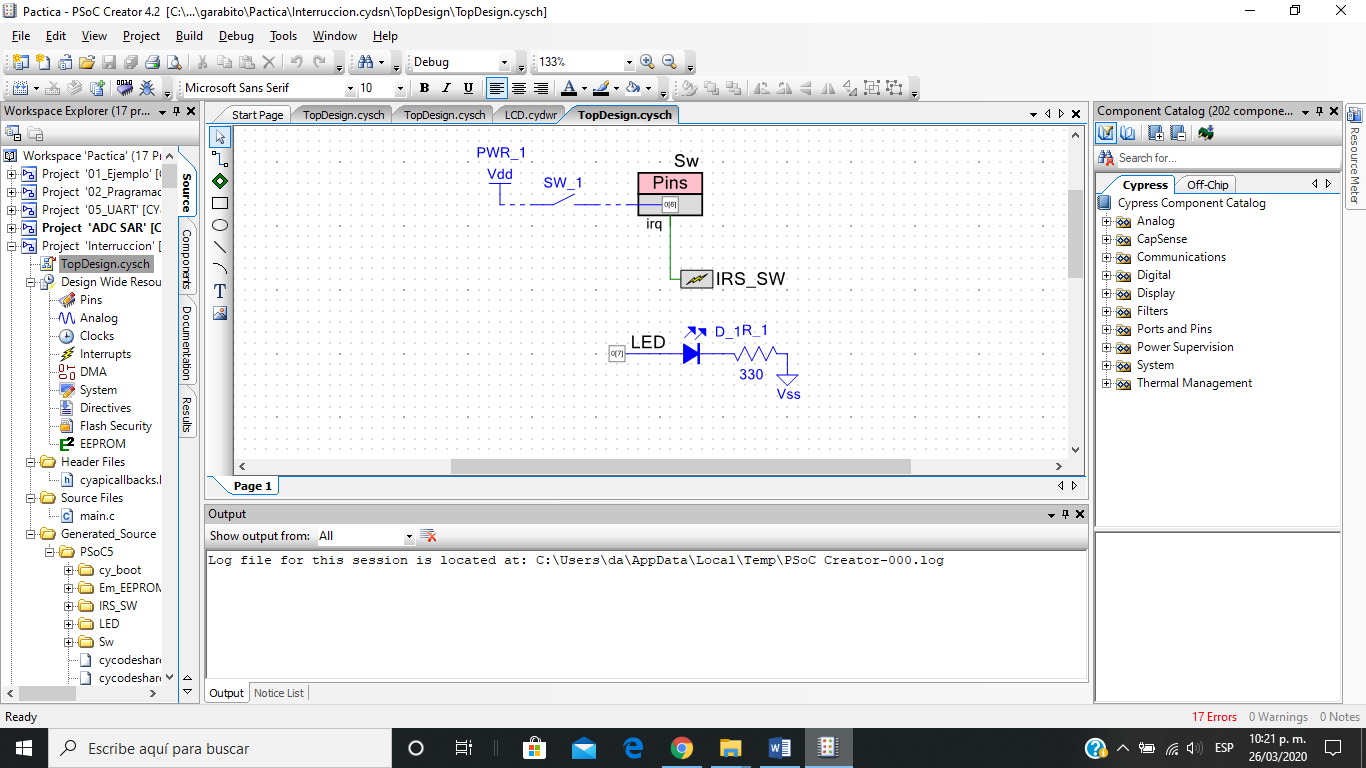
t\_interrupcion = (4 x pre-escala x 256 / fosc = (4 x 1 x 256) / 8000000 = 128 us

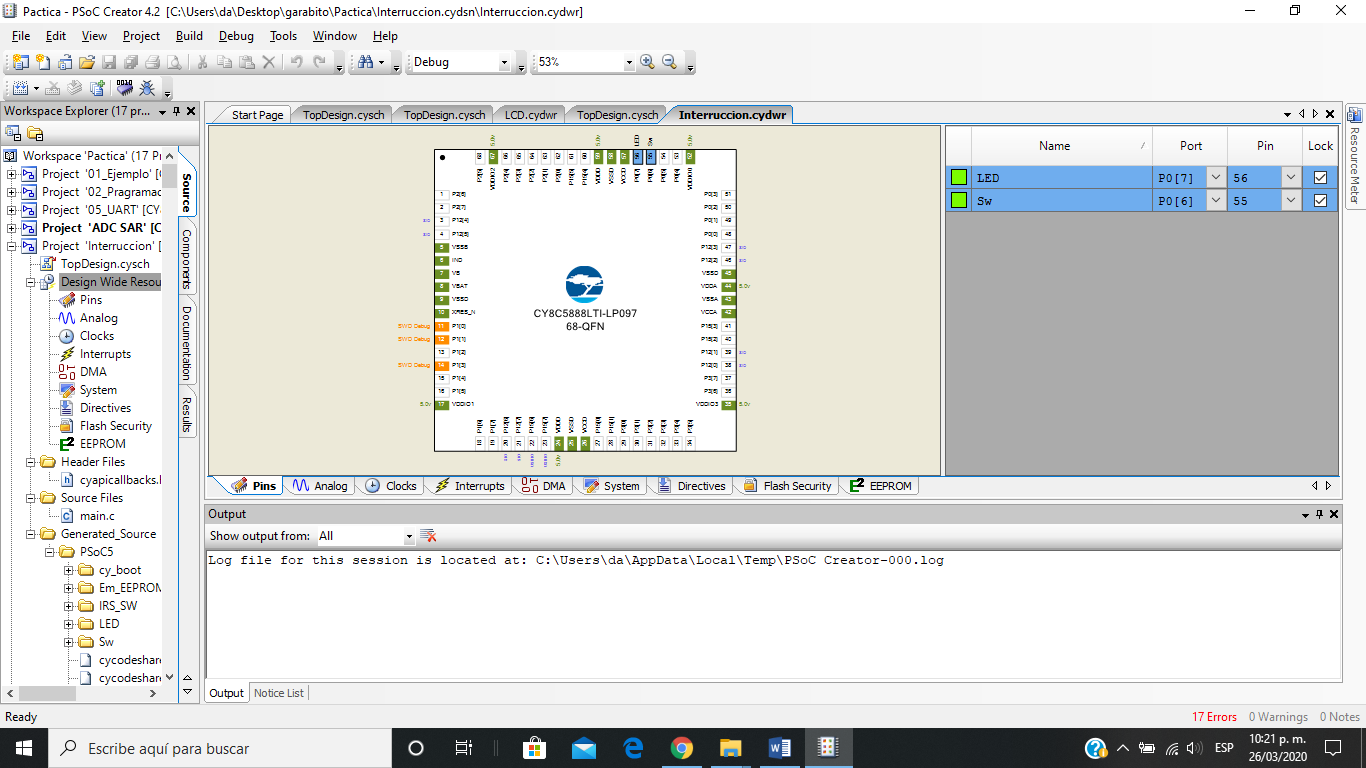
Con este valor, el parpadeo del LED se puede calcular del siguiente modo:

T = 2 x t\_interrupcion x MAX\_LED\_CNT = 2 x 128 us x 2000 = 512 ms

Utilizando un registro sombra que actualiza GPIO periódicamente en la rutina principal y recordando que “main.h” es el mismo de las secciones anteriores,

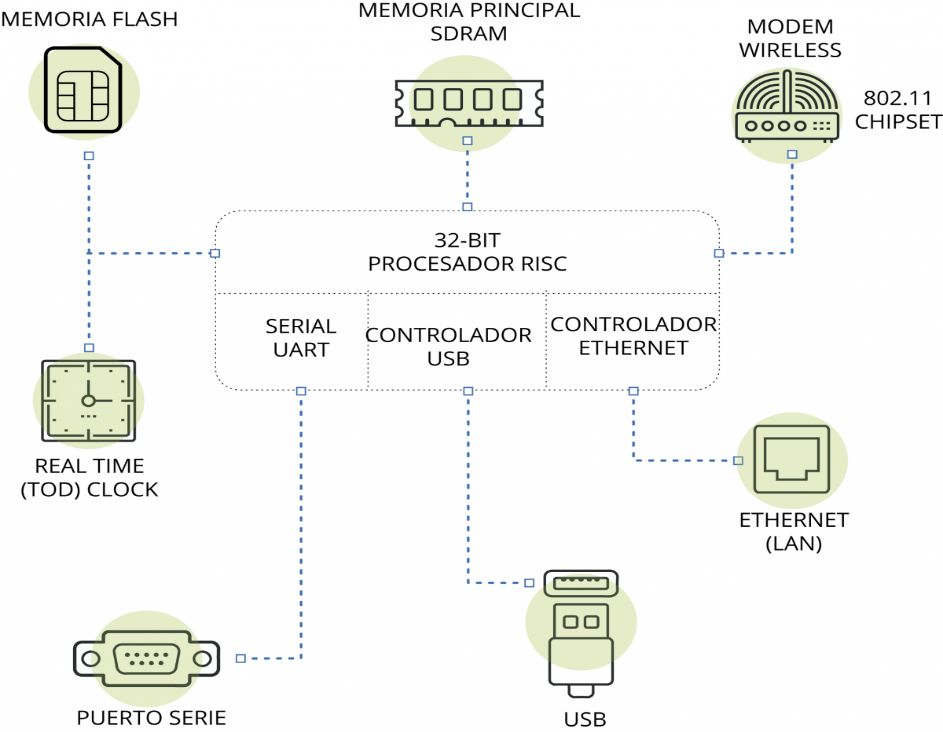
**(Top desing)**





**(Imagen de muestra)**





**(Código)**

/\* ========================================

\*

\* Copyright YOUR COMPANY, THE YEAR

\* All Rights Reserved

\* UNPUBLISHED, LICENSED SOFTWARE.

\*

\* CONFIDENTIAL AND PROPRIETARY INFORMATION

\* WHICH IS THE PROPERTY OF your company.

\*

\* ========================================

\*/

#include "project.h"

//CY\_ISR(interruptor)

//{

//LED\_Write(~LED\_Read());

//CyDelay(100);

//Sw\_ClearInterrupt();

//}

CY\_ISR(algo)

{

LED\_Write(~LED\_Read());

Sw\_ClearInterrupt();

}

int main(void)

{

CyGlobalIntEnable; /\* Enable global interrupts. \*/

IRS\_SW\_StartEx(algo);

/\* Place your initialization/startup code here (e.g. MyInst\_Start()) \*/

for(;;)

{

/\* Place your application code here. \*/

}

}

/\* [] END OF FILE \*/