Parte 1

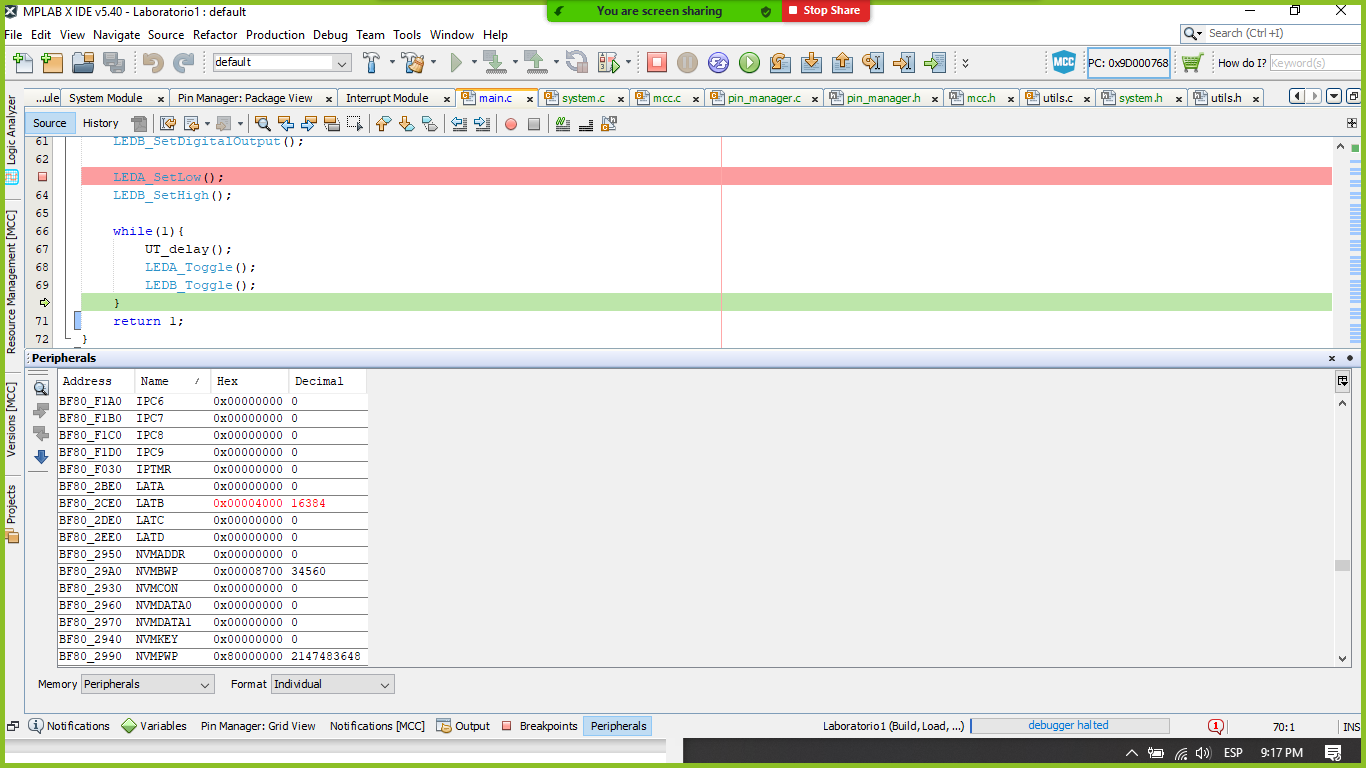
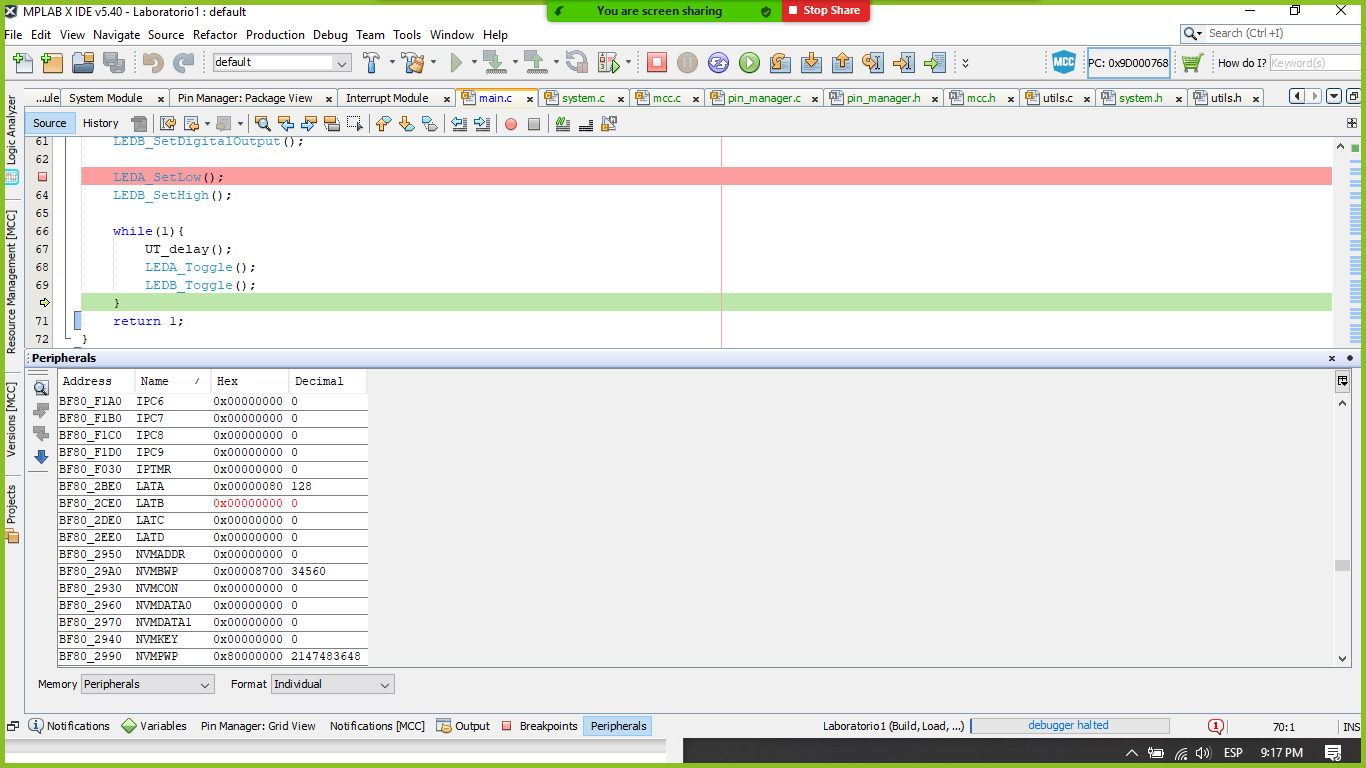
1.g

El archivo pin\_manager.h se puede relacionar con el archivo HardwareProfile.h debido a que en el mismo se declaran y definen las funciones relacionadas a los LEDs.

Como en el pin\_manager.h las funciones tienen los mismos nombres que en HardwareProfile.h, este archivo ya no será necesario incluirlo en el proyecto.

Con respecto a los otros archivos generados en el MCC, estos corresponden a las configuraciones que previamente se realizaban en el main.c, antes de iniciar el programa.

1.f

2.a

Debemos configurar los botones, que se encuentran en los pines RB15 y RA13 (S2 y S3 respectivamente). Los mismos se deben configurar como pines de entrada.

2.b

Por lo visto en el circuito esquemático de la placa de desarrollo, se necesita una resistencia de pull-down para que coincida con el simulador, ya que se verifico, que cuando no está presionado el botón hay una entrada lógica de 0.

2.d

Los macros que se generar son las siguientes:

BTNX\_SetHigh: Para darle el valor de 1 al botón X

BTNX\_SetLow: Para darle el valor de 0 al botón X

BTNX\_Toggle: Cambiar el valor actual del botón X.

BTNX\_GetValue: Indica el valor del botón X.

BTNX\_SetDigitalInput: Configura al botón X como pin de entrada

BTNX\_SetDigitalOutput: Configura al botón X como pin de salida

Mientras que las funciones que se generan son BTNX\_SetValue.

Parte 2

1.c

El código agregado fue el siguiente:

BTNX\_CallBack

BTNX\_SetInterruptHandler

BTNX\_SetIOCInterruptHandler

Los registros configurados para las interrupciones de los pines de los botones son:

CNIE0A13, CNIE0B15, CNFA13, CNFB15, CNAIE, CNBIE, CNAIF, CNBIF

Segunda Parte

2.a //Hay que modificarlo

Cambiaron los siguientes archivos:

Interrupt\_manager.c

mcc.h

system.c

tmr2.c

tmr2.h