***Laboratorio 5 – Sensor de Temperatura LM35***

Objetivos del laboratorio son conocer:

* Utilizar la UART para enviar
* Convertidores Analógicos a Digitales (ADC)
* Sensar temperatura

1. Sensado de temperatura

*1.1 – Introducción*

En este apartado utilizaremos sensores de temperatura. Existen de varios tipos, prácticamente dos, uno será analógico y otro digital. Con analógico nos referimos a que la salida del dispositivo es un rango de voltaje y con digital que la salida de voltaje es un protocolo de comunicación, ya sea I2C o SPI. Ambos tienen la misma función (medir temperatura) sin embargo el último habrá que configurar registros antes de leer el registro final que conlleva a la medición de la variable temperatura.

Para procesar este dato analógico necesitamos un elemento que transforme esta variable de salida (voltaje) a digital y para esto utilizaremos un ADC por medio del XADC de Xilinix que nos permitirá unificarlo en un elemento digital como el Microblaze.

El XADC está insertado en el núcleo del Arty (en el Artix-7) y nos ayudará a realizar esta función, existen varios canales analógicos, no solo uno, sino que podemos tener varios elementos analógicos a sensar.

*1.2 – Introducción al LM35*

El LM35 es un sensor de temperatura común, que puede conseguirse en cualquier electrónica local. Este tiene un rango dinámico de voltaje de entrada de 4 a 20V. Localmente se puede conseguir uno de encapsulado TO-92(3).

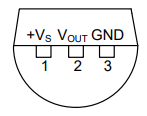
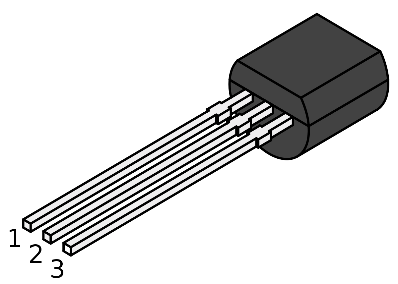


Figura 1.1. Encapsulado TO-92(3) del LM35. Descripción de pines, donde +Vs es el voltaje de alimentación y Vout es el voltaje analógico de salida a medir.

Es importante también considerar la característica del voltaje de salida, es decir, como cambia la temperatura con respecto al voltaje para nosotros poder interpretar la variable. El LM35 exhibe un caso lineal como lo dice la hoja de datos; otros sensores de temperatura como el TMP35 habrá que revisar la gráfica y obtener la ecuación de linealización del sensor.

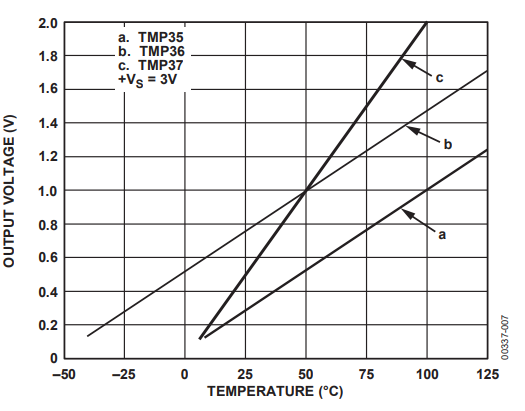


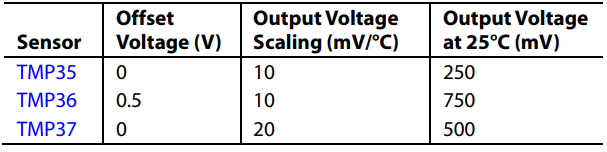
Figura 1.2. Gráfico del TMP35. El gráfico es lineal pero con una pendiente diferente para cada sensor, este es el caso de otro sensor, el TMP35, utilizará la ecuación lineal de ‘a’.

Para el caso del LM35 tenemos una ecuación de salida:

Ecuación 1.1. Cálculo del voltaje de salida del LM35.

En el caso del TMP35, en la tabla 4 de la hoja de datos se exhibe la siguiente tabla.

Tabla 1.1. Valores de offset, escalamiento y valor de salida del TMP35.



Utilizaremos esta tabla para explicar la linealización de sensores.

1.3 – Linealización de Sensores (ejemplo para el TMP35)

Recordando nuestras clases de ecuaciones tenemos la ecuación de la línea recta tenemos

Ecuación 1.2. Ecuación de la línea recta.

‘m’ es la pendiente o la tasa de cambio de la temperatura (10 mV/°C para ambos casos TMP35 o LM35)

‘y’ es el voltaje de salida del sensor (Vout)

‘x’ es la temperatura que se está captando en grados celcius

‘b’ es la intersección de la gráfica con el eje de temperatura, lo que se llama “offset”

Así, los parámetros, para el TMP35 son:

b = 500mV, m = 10 mV/°C, x = temp\_c, y = voltage\_mv

Si deseamos entonces de la ecuación de la línea recta calcular la temperatura tenemos la siguiente ecuación:

Ecuación 1.3. Cálculo de temperatura para el TMP35. En el caso del LM35 b = 0.

1.4 – Resolución del sensor

En el caso de los ADC, es importante la resolución. En la Arty tenemos un ADC de 12 bits, lo que conlleva a 212 = 4096 posibilidades. En el caso, se calcula la resolución de:

Ecuación 1.4. Cálculo de resolución. Por ejemplo, si se alimenta el ADC con un voltaje de 3.3V vs 5V.

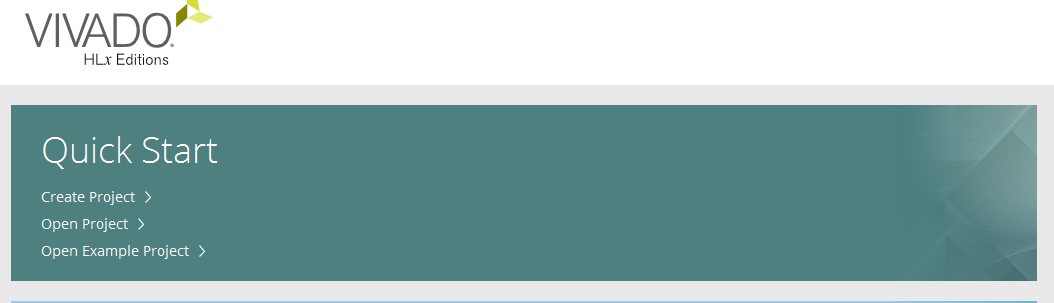
En el caso del ADC notamos que a 3.3V tenemos una resolución mucho mayor al del sensor, es decir 0.8 mV para el ADC de la Arty vs 10 mV por cambio de °C, lo que nos dá un mayor rángo dinámico para detectar cambios. Si el ADC se alimentase de 5V el rango sería menor como se observa en el cálculo.

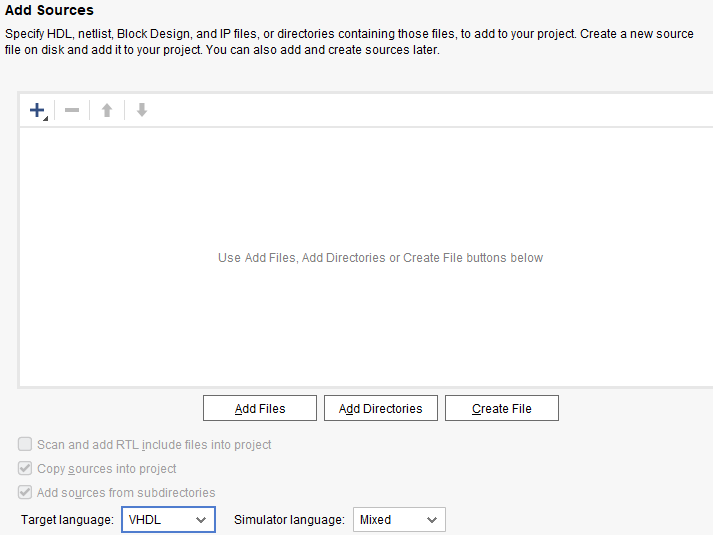
2. – Creación del Proyecto en VIVADO

2.1 – Creación del Proyecto de Sensado de Temperatura

***NOTA: FAVOR CREAR EL PROYECTO EN UNA UBICACIÓN DONDE NO HAYA ESPACIOS, DE LO CONTRARIO EL XILINX SDK CREARÁ EL PROYECTO FUERA DE LA LOCALIDAD DEL PROYECTO.***

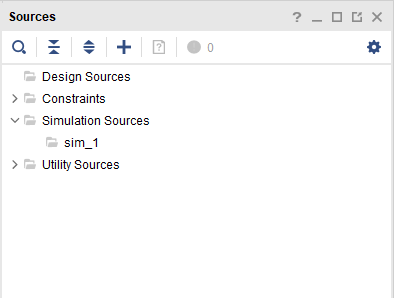
2.2 - Abrir Vivado en la pantalla principal y presionar en abrir proyecto





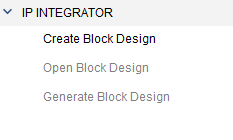
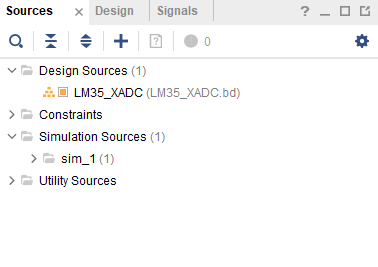
*Figura 2.1 – Creación del proyecto en Vivado. Asegurese de elegir VHDL!.*

2.3 – Crear del Project manager, un proyecto en blanco:



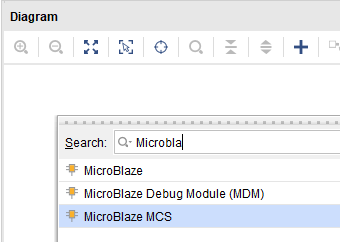
*Figura 2.2 – Archivos de trabajo de proyecto*

2.4 – De la ventana de Flow Navigator a su izquierda cree un diagrama de bloques de la sensado de temperatura.

*Figura 2.3 – Creación de diagrama de bloques*

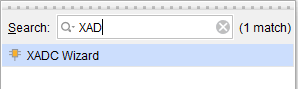
2.5 – Del diagrama de bloques anterior creado, inserte el bloque de Microblaze MCS presionando ‘+’ y buscando el bloque Microblaze MCS que es el Microblaze de menor capacidad.



*Figura 2.4 – Inserción de Microblaze al proyecto*

2.6 – Podemos utilizar el Microblaze normal, pero tomaría muchos componentes lógicos, si hace doble click verá la siguiente ventana.

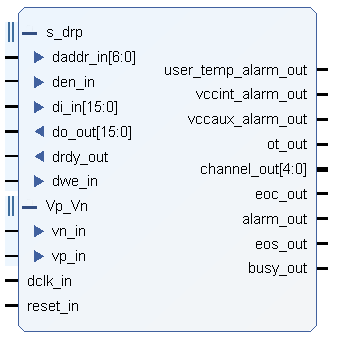
2.7 – Agregar también el XADC wizard para poder utilizar el convertidor analógico a digital.



*Figura 2.5 – XADC. Convertidor analógico a digital.*

2.8 – Presione doble click para configurarlo.

Para configurar el ADC vaya a la pestaña ‘Basic’ y seleccione DRP (Dynamic Reconfigurable Port). Si expandimos el icono de ‘+’ del diagrama de bloques de configuración del XADC observaremos que el daddr\_in es persistente (aparece) y podremos seleccionar el canal ADC que queremos medir. La salida del XADC será captada del do\_out[15:0] en un valor digital.

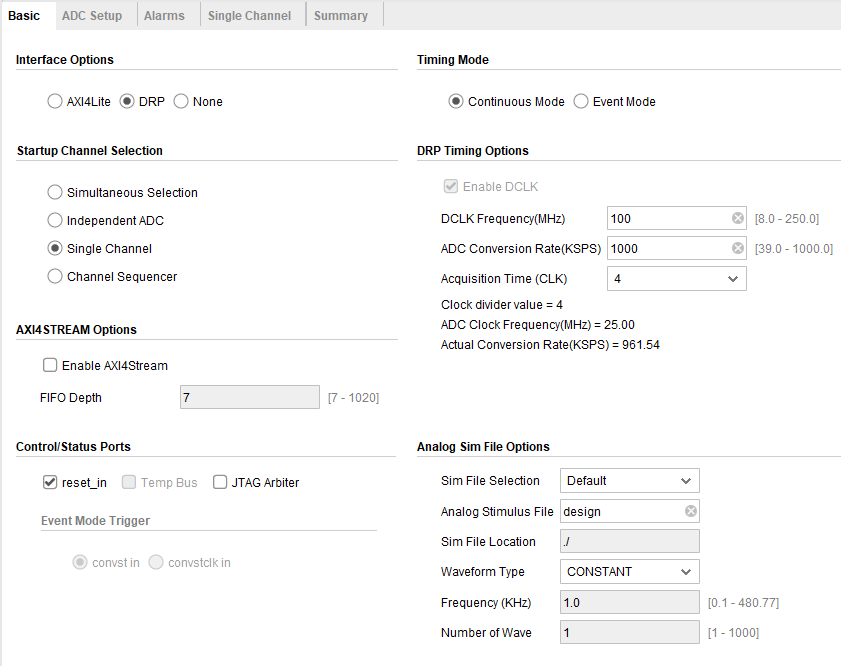


*Figura 2.5 – XADC Expandido. Note la entrada de dirección daddr\_in (canal a multiplexar) y do\_out[15:0] que posee el valor de conversión de salida.*

2.9 – Seguido, configurar en la pestaña Basic que necesitaremos.

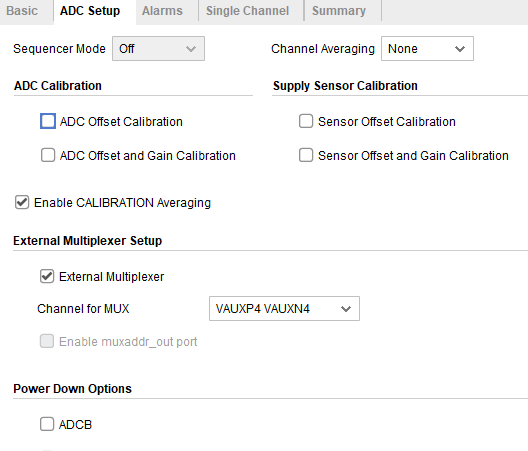
Configuraremos para ingresar un solo canal de conversión ‘Single Channel’, utilizaremos la entrada de reset del ADC ‘reset\_in’ la cual conectaremos al reinicio del sistema. El modo de conversión será de modo continuo ‘Continuous Mode’ y las opciones de tiempo del reloj son de 100 MHz debido al reloj de entrada del sistema, las muestras por segundo serán de 1000 KSPS (Kilo Samples Per Second) sin embargo observamos que el máximo que obtendremos será de 961.54 lo cual basta y sobra para esta aplicación. El tiempo de adquisición será de un factor de 4 y el ADC tendrá un reloj de entrada de 25 MHz.

Todo lo anterior se puede observar en la siguiente figura.



*Figura 2.6 – Opciones de XADC (Basic). Ajustar el XADC para realizar las mediciones.*

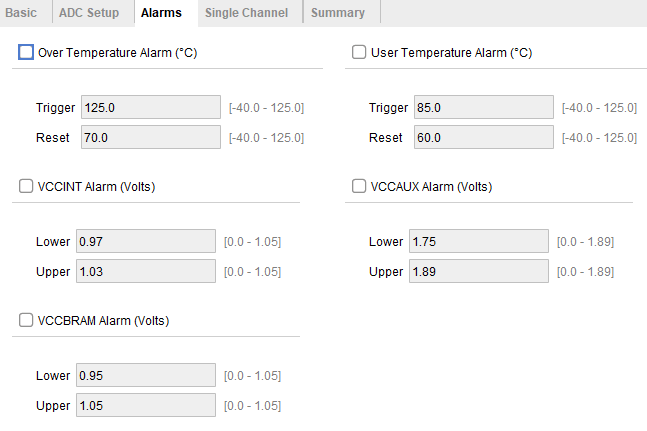
Ajustar la ventana de ADC Setup para como se muestra en la siguiente figura:



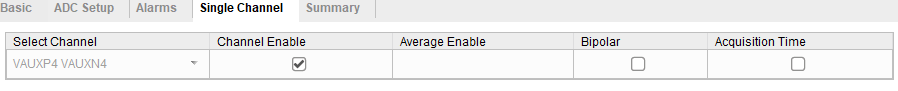
*Figura 2.7 – Opciones de XADC (ADC Setup). No utilizaremos ganancias u offsets, con lo más básico es simple y necesario para esta aplicación.*

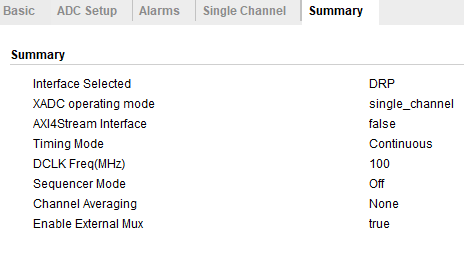
Seleccionaremos un canalo de MUX, el VAUXP4 VAUXN4, debido a que utilizaremos el Pin A0.

No utilizaremos ninguna alarma, el Arty tiene la capacidad e informarnos si dentro del encapsulado ocurre alguna anormalidad, tiene algunos arreglos de diodos que miden temperatura y alertan alguna mala operación del sistema.



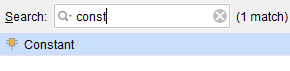
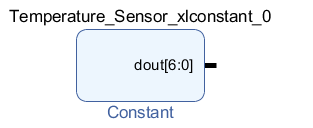
*Figura 2.8 – Opciones de XADC (Alarms). No utilizaremos alarmas*





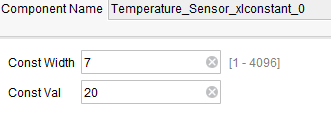
*Figura 2.9 – Ventanas de Single Channel y Summary. Resumen de la configuración del XADC.*

2.10 – Agregue una constante para seleccionar el canal ADC

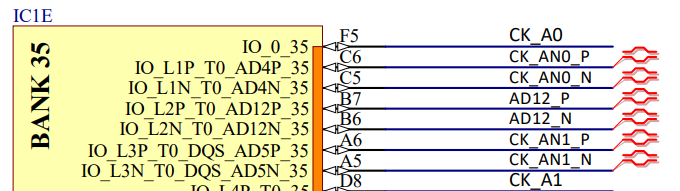
 

*Figura 2.10 – Agregar una constante. Esta constante es para fijar el canal de medición en A0.*

2.11 – Configure la constante para un valor de 20 y un ancho de 7



*Figura 2.11 – Ventana de Configuración de constante. Configurar el valor de 20 para el VAUXP4 (A0).*



*Figura 2.11 – Esquemático de Arty A7. Observamos que el pin A0 está al AD4P y AD4N*

Revisando un poco más la configuración del Arty-A7, en el manual de XADC

<https://www.xilinx.com/support/documentation/user_guides/ug480_7Series_XADC.pdf>

Observamos que en la página 38 tenemos.

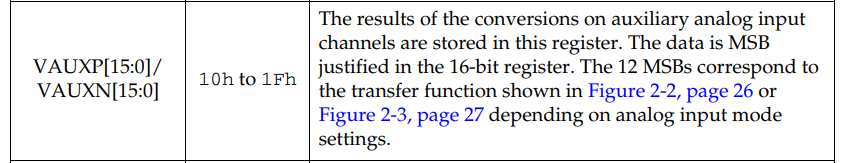
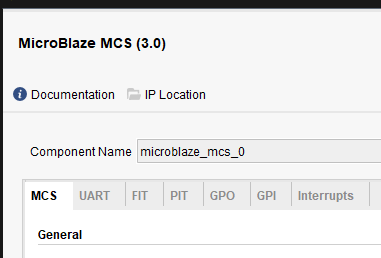
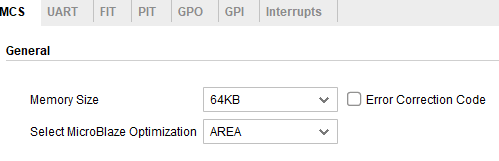


Figura 2.12 – El XADC, según el manual, el VAUX4P y VAUX4N quedan en la posición hexadecimal 0x14 que convertido a decimal será 20.

2.12 – Configurar microblaze como se muestra en las siguientes figuras.

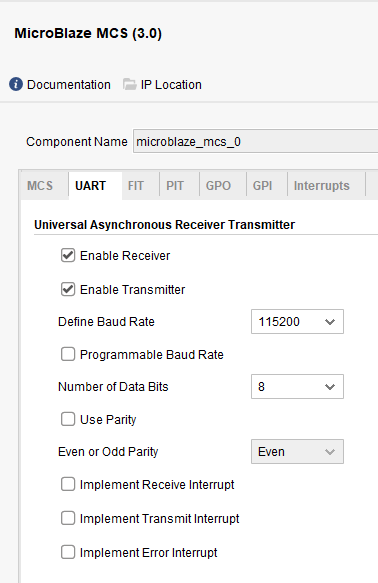


*Figura 2.13 – Ventana general de Microblaze MCS.*



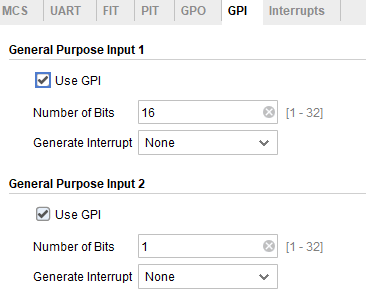
*Figura 2.14 – Ventana de Microblaze MCS, habilitar 64Kb de memoria de bloque.*

2.13 – Para la sección de UART tendremos entonces que habilitar tanto el transmisor como el receptor.



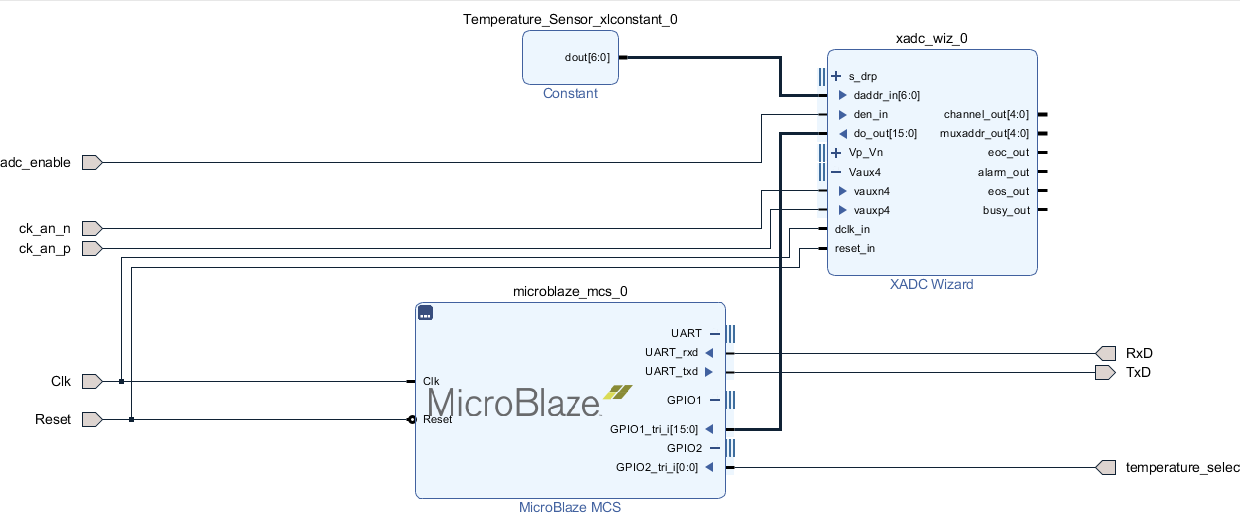
*Figura 2.15 – Habilitando la UART.*

2.14 – Para la sección de GPI habilitar las entradas de adquisición digital del XADC al GPIO y un pin general para leer de Fahrenheit o Celcius.



*Figura 2.16 – Habilitar los 16 registrosd el XADC y un bit para cambiar de farenheit a celsius.*

2.15 – Conectar todos los elementos como se muestra en la figura.



*Figura 2.17 – Conexión de todos los componentes Microblaze MCS, XADC, Señales de I/O y constante.*

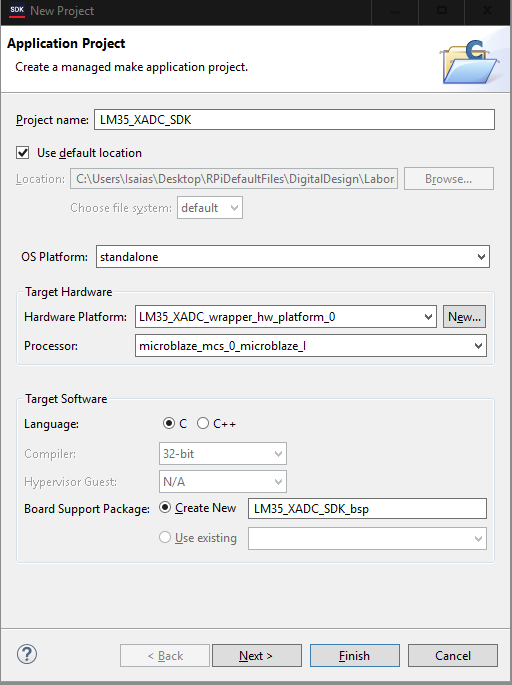
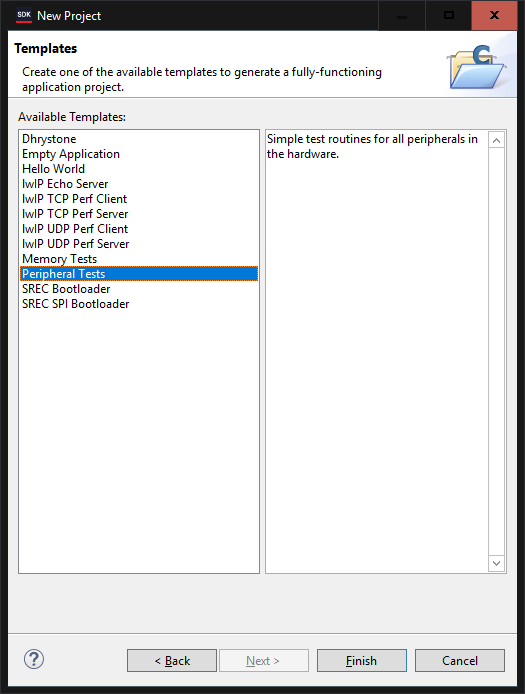
3 – Interfaz en XSDK al Xilinx XADC

*3.1 - Creación del Proyecto de Sensado de temperatura en XSDK*

Para la siguiente sección debe tener listo:

* Creado el HDL wrapper como el laboratorio anterior
* Haber generado la síntesis e implementación del diseño del HDL Wrapper
* Haber exportado el hardware

3.2 – Crear un nuevo proyecto, al hacerlo seleccionar la plantilla de ‘Peripherals Tests’

*Figura 3.1 – Ventana de Nuevo Proyecto de Hardware. Crear un nuevo proyecto de hardware para el XADC, utilizar la plantilla Peripheral Tests.*

3.3 – Creado el proyecto añada el siguiente código que tomará el dato del ADC y lo enviará por serial.

*Listado 3.1 – Código de lectura de convertidor ADC y envío de datos seriales a la UART para despliegue en la terminal de la PC.*

#include <stdio.h>

#include "xparameters.h"

#include "xil\_cache.h"

#include "xiomodule.h"

int main()

{

// Parameters

u32 data, ADC\_reading, Selected\_Degree;

float Voltage\_mV, Temperature\_C, Temperature\_F;

float Temperature\_out;

int delay\_count, Temperature\_whole, Temperature\_hundreds;

XIOModule gpi;

// initialize our GPIO's

data = XIOModule\_Initialize(&gpi, XPAR\_IOMODULE\_0\_DEVICE\_ID);

data = XIOModule\_Start(&gpi);

while(1)

{

ADC\_reading = XIOModule\_DiscreteRead(&gpi, 1); // Perform ADC Reading

ADC\_reading = ADC\_reading >> 4; // shift 4 LSB's

Selected\_Degree = XIOModule\_DiscreteRead(&gpi, 2); // Read the slide switch to determine C or F

Voltage\_mV = (float)(3.3/4095) \* (float)ADC\_reading \* 1000; // determine voltage reading

Temperature\_C = (Voltage\_mV - 0) / 10; // Calculates temperature in C

Temperature\_out = Temperature\_C;

if (Selected\_Degree == 1)

{

// Calculate the temperature in farenheight

Temperature\_F = (Temperature\_C \* 1.8) + 32;

Temperature\_out = Temperature\_F;

}

// Write the Output Temperature

Temperature\_whole = Temperature\_out;

Temperature\_hundreds = (Temperature\_out - Temperature\_whole) \* 100;

if (Selected\_Degree == 0)

xil\_printf("Temperature: %d.%d C\n\r", Temperature\_whole, Temperature\_hundreds);

else

xil\_printf("Temperature: %d.%d F\n\r", Temperature\_whole, Temperature\_hundreds);

// delay

delay\_count = 0;

while (delay\_count < 50000000)

delay\_count++;

}

//Xil\_ICacheEnable();

//Xil\_DCacheEnable();

// print("---Entering main---\n\r");

// print("---Exiting main---\n\r");

//Xil\_DCacheDisable();

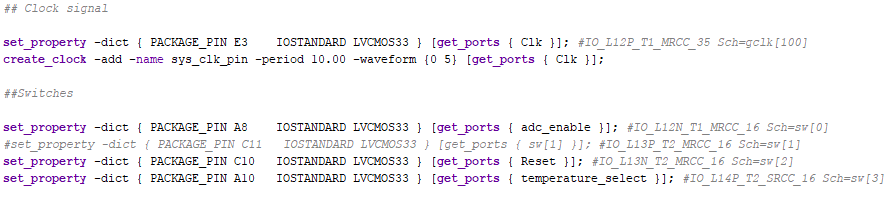
//Xil\_ICacheDisable();

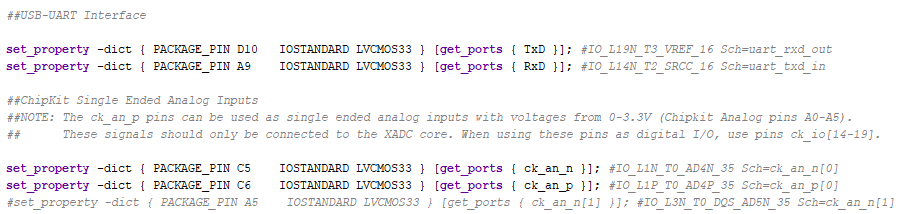
return 0;

}

3.4 – Compile el programa anterior

3.5 – Retorne a Vivado e Inserte el archivo XDC como en laboratorios anteriores y cambie el siguiente contenido.





*Figura 2.6 – Adición y modificación del archivo maestro Arty\_A7\_Master.xdc.*

3.6 – Guardar la modificación del archivo .XDC y seguidamente asociar el archivo .elf como en laboratorios anteriores.

3.7 – Generar el bitstream y quemar en la tarjeta el programa.

4 – CoolTerm e Interfaz a la PC

NOTA: Seguir el laboratorio 6 en la sección 5 si tiene dudas de iniciar coolterm.

4.1 – Inicie CoolTerm bajo la interfaz configurada en microblaze, esto es: 115200,8,N,1 y sin control de flujo.

4.2 – Configurado CoolTerm y detectado el puerto de hardware COM en la PC, abrir el puerto de comunicaciones.

4.3 – Conecte según la siguiente table el LM35 al Arty.

*Tabla 4.1 – Conexión del Arty A7 y LM35*

|  |  |
| --- | --- |
| Arty | LM35 |
| 3V3 | 1 |
| A0 | 2 |
| GND | 3 |

***NOTA: Se está conectando el A0 directamente al sensor LM35, sin embargo la salida de este sensor aumentará a altas temperaturas sobre los 3.3V, considerar esto si se va a sensar un valor analógico muy alto, para este laboratorio no existen estos problemas. Si tiene la oportunidad, considere el TMP35 que funciona directamente a 3.3V de salida máxima.***

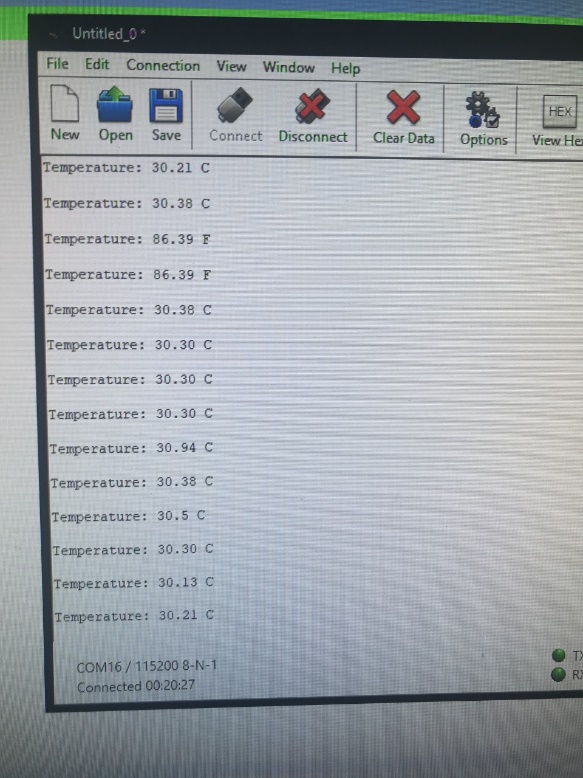
4.4 – Considere la siguiente tabla para que funcione la aplicación:

*Tabla 4.2 – Estado de E/S para función de la aplicación. N/A = No Aplica*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Switch | Pin en .XDC | Descripción | Estado para Función Normal | UP (Arriba) | DN (Abajo) |
| SW0 | A8 | Habilita el ADC | UP | Habilita el ADC | Deshabilita ADC |
| SW1 | C11 | N/A | N/A | N/A | N/A |
| SW2 | C10 | Reset | UP | UP = Arranca Microblaze | DN = Desactiva Microblaze |
| SW3 | A10 | Selecciona C o F | DN | Farenheit | Celcius |

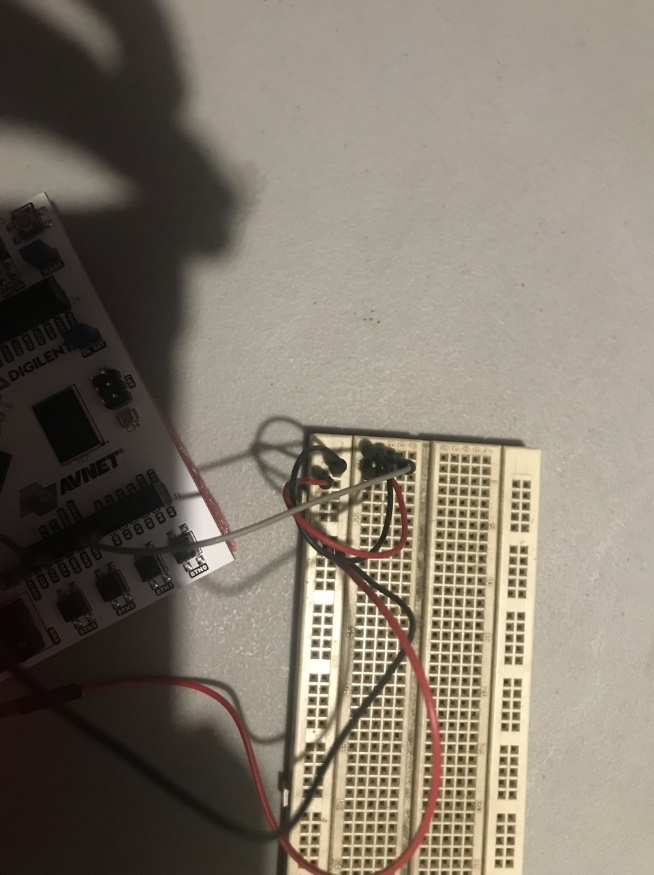
4.5 – Si aún no ha descargado el bitstream, descárguelo a la tarjeta.

4.6 – Observe en la terminal CoolTerm, debe ver una salida como sigue:

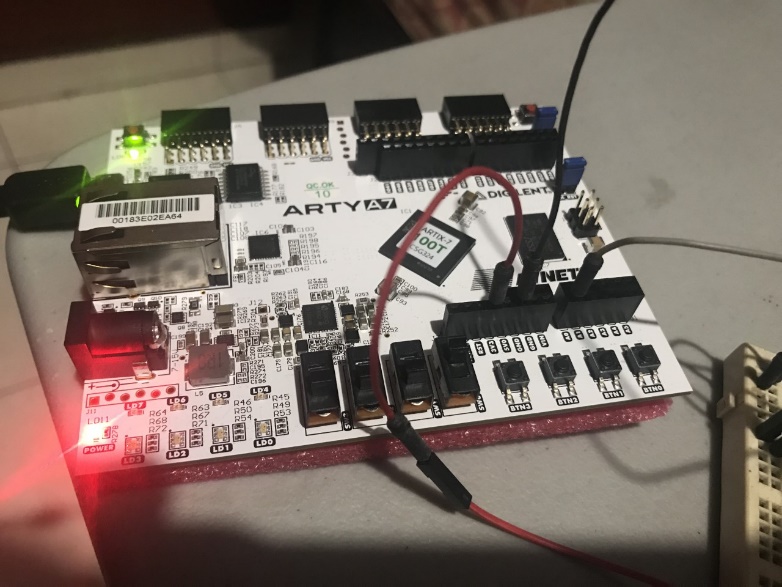


*Figura 4.1 – Vistazo final a la aplicación de comunicación del puerto serial.*

4.7 – Debajo se muestran imágenes del proyecto



*Figura 4.2 – Sensor de Temperatura LM35 en protoboard.*



*Figura 4.3 – Conexiones de Sensor de Temperatura en la Arty-A7.*

5 - Evaluación

* 50% - Terminar el proyecto anterior
* 50% - Modificar los baudios según la tabla dada a continuación y demostrar en el FPGA

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ADC\_Channel | ADC\_Enable | ADC\_Enable | Reset (SW) | Baudrate |
| 4 | 0 | 3 | 1 | 38400 |
| 2 | 0 | 2 | 1 | 19200 |
| 5 | 2 | 3 | 0 | 900 |
| 0 | 1 | 0 | 2 | 115200 |
| 5 | 3 | 1 | 2 | 300 |
| 3 | 3 | 0 | 1 | 600 |
| 2 | 2 | 3 | 0 | 57600 |
| 3 | 0 | 3 | 1 | 38400 |
| 3 | 1 | 2 | 0 | 900 |
| 3 | 3 | 0 | 2 | 1200 |
| 0 | 3 | 1 | 0 | 19200 |
| 5 | 2 | 3 | 1 | 600 |
| 5 | 0 | 1 | 2 | 57600 |
| 4 | 2 | 3 | 0 | 600 |
| 2 | 0 | 1 | 3 | 115200 |
| 5 | 3 | 2 | 0 | 57600 |
| 3 | 1 | 0 | 3 | 900 |