

# Guia para la configuración de la tarjeta zybo z10, para la implementación del pmod AD5.

## Sistema de lectura secuencial de 4 u 8 canales diferenciales

Gonzalo Burgos Fuentes<sup>a</sup>

<sup>a</sup>CePIA, Departamento de Astronomía, Facultad de Ciencias Físicas y Matemáticas, Universidad de Concepción

Compiled on January 28, 2020

### 1. resumen

En este documento se presentarán las herramientas y fuentes necesarias para aprender como configurar una tarjeta de desarrollo, junto al uso de su procesador. El objetivo es entregar una guía para entender la programación de tarjetas de desarrollo basadas en FPGA y en particular programar el modelo de tarjeta Zybo\_z7-10. Exponiendo los resultados obtenidos, a partir de la configuración de lectura pseudo-diferencial y utilizando los 8 canales de forma secuencial.

### 2. Objetivos

Este trabajo responde a la necesidad del proyecto [A Pseudo-Correlation MMIC-based 183 GHz Water Vapor Radiometer\\*](#), donde se requiere la lectura y monitoreo de las corriente, que energiza la tarjeta PCB desarrollada para el control y protección de los módulos de JPL implementados en el "radiometer" y donde se estima que los valores de lectura variaran entre los mili y micro amperes.

#### 2.1. Objetivo principal

- Implementación de la lectura secuencial de 8 canales diferenciales desde el pmod AD5, a través del procesador de la tarjeta Zybo z7-10.

#### 2.2. Objetivo secundarios

- Diseño de arquitectura digital, para la implementación en hardware.
- Configuración y programación del hardware para la toma de datos.
- Lectura y visualización de datos.

### 3. Hardware y software

#### 3.1. Hardware

El hardware ha implementar esta dividido en 2 elementos, el primero corresponde a la tarjeta de desarrollo Zybo z7-10, sus especificaciones y características se encuentran en disponibles en la pagina de Digilent, a la cual puedes acceder directamente haciendo click[aquí](#) y el Pmod AD5 cuya información se encuentra también disponible en la pagina de Digilent y a la cual puedes acceder directamente haciendo click [aquí](#)

#### Tarjeta Zybo-z7-10

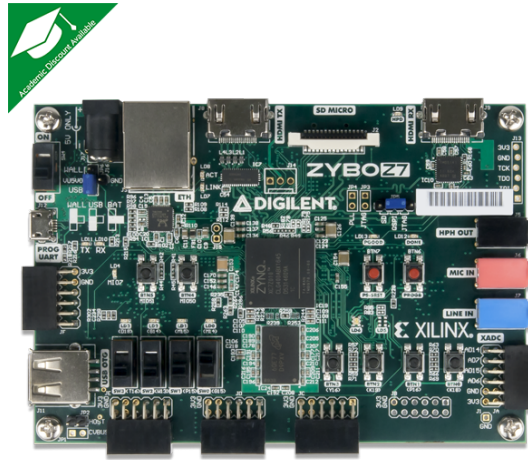
Para el desarrollo de nuestro objetivo las características indispensables a utilizar de esta tarjeta son sus puertos pmod, su procesador<sup>†</sup> y Field-programmable gate array (FPGA)<sup>‡</sup>. Ya que permite implementar de forma independiente la configuración del procesador y la arquitectura FPGA. la cual a diferencia de otros modelos no utilizan sus recursos lógicos de la FPGA en sintetizar un procesador. En la figura 1, se muestra una imagen de la tarjeta, las especificaciones y funciones de cada uno de sus componentes se encuentra disponible en el pagina del proveedor <sup>§</sup> y esta imagen es de carácter ilustrativa para el lector.

\* Para mayor información sobre este proyecto, pueden acudir directamente al paper o solicitar exceso a toda la información a la líder del proyecto Katherine Cortés

† La programación esta arquitectura esta basa en el lenguaje C, Por lo que os recomiendo tomar un curso o leer sobre la sintaxis de este lenguaje antes de empezar.

‡ Para mayor información, métodos y lenguajes de programación de esta arquitectura, puedes buscar en la web. Pero os recomiendo leer las notas o tomar el curso "DASD" del profesor Miguel Figueroa, del Departamento de Ingeniería Eléctrica de la Universidad de Concepción

§ Link disponible [aquí](#)

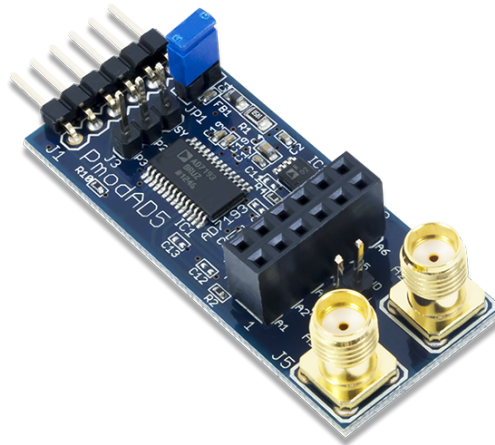


**Figure 1.** Imagen de la tarjeta Zybo z7-10, extraída directamente de la pagina del distribuidor Digilent.

### Pmod AD5

Los "Peripheral Module interface" han sido diseñados para simplificar la integración de distintos módulos y/o extensiones desde o hacia las tarjetas de desarrollo. Una completa descripción de su funcionamiento y arquitecturas se encuentra disponible en la pagina de Digilent, a la cual puedes acceder directamente haciendo click [aquí](#) y donde además se encuentra la información sobre los distintos protocolos de comunicación. En la figura 2, se muestra una imagen del pmod AD5 el cual a sido seleccionado tanto por el numero de bit de conversión (24 bits), rango conversión ( $\pm 2.5V$ )<sup>¶</sup> y su número de canales (8).

El principal componente de pmod AD5 corresponde al "AD7193", su funcionamiento y configuración se encuentran completamente descrito y especificado en su datasheet "AD7193", disponible en la pagina del fabricante Analog Devices, la cual puedes acceder directamente haciendo click [aquí](#)<sup>||</sup>.



**Figure 2.** Imagen del Pmod AD5, extraída directamente de la pagina del distribuidor Digilent.

<sup>¶</sup> Resolución de conversión

<sup>||</sup> Para entender la configuración de este elemento sin tener conocimientos previos es esencial leer este datasheet y los protocolos de comunicación

## 3.2. Software

Para la configuración y control del hardware se utilizarán 2 software; el primero de ellos es *python*, el cual se encarga de configurar la conexión de lectura de datos desde la tarjeta y su visualización. y *Vivado*, el cual diseña, configura y controla el hardware que es implementado en la tarjeta de desarrollo.

### Python

Python un lenguaje de programación interpretado de alto nivel y de propósito general, creado por Guido van Rossum y lanzado en 1991, este permite integración de diferentes funcionalidades mediante el uso de paquetes adicionales a su sintaxis básica. Transformándose así en un de los lenguajes mas versátiles y de uso libre disponibles en la web, para revisar su documentación básica puedes dirigirte directamente a la web de los desarrolladores (<https://www.python.org/>) y aprender la sintaxis desde las bases mismas de sus desarrolladores, o si ya posees conocimientos en programación puedes hacer click [aquí](#) para a un breve curso web sobre python, el cual te dará un panorama general (recordatorio) del lenguaje y sus aplicaciones.

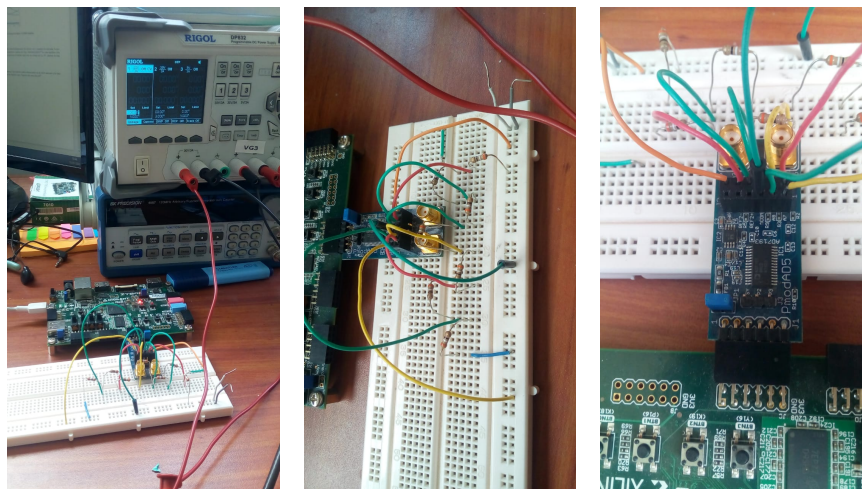
### Vivado

Vivado es un software desarrollado por Xilinx para la síntesis y análisis del lenguaje descriptivo de hardware o HDL\*\* por sus siglas en ingles (Hardware Description Language), el cual es utilizado para describir la estructura y comportamiento de los circuitos eléctricos de un sistema, permitiendo implementar de manera sencilla lógica digital. La mayor diferencia entre este tipo de lenguajes y los de mayor nivel como python, ya se en la consideración del tiempo o reloj interno al que funciona el hardware y por lo tanto los ciclos en los que se realizaran cada una de las sentencias lógicas programadas. Una completa descripción del software se encuentra disponible en la pagina de de xilinx (<https://www.xilinx.com/>), otro enlace que resulta útil para aprender sobre este software y la implementación en hardware a través de la herramienta SDK de vivado es el siguiente [link](#) donde se encuentra explicito en la configuración y uso del Pmod, a través del procesador.

## 4. Setup

Para configurar y comprobar el funcionamiento del sistema, es necesario generar un setup que permita leer simultáneamente los 8 canales del digitalizador. Para ello se configuró un circuito sencillo sobre un protoboard, el cual cuenta con 7 resistencias de 330 ohms. Los cuales actúan como divisores de tensión, que en la practica generan 8 nodos para la lectura de voltaje, dependiendo del punto al que sea conectado. En caso de no poseer conocimientos previos sobre este tipo de configuración, ingresar al siguiente [link](#) donde podrás encontrar un ejemplo de ello y la implementación en arduino.

El circuito implementado en la protoboard es conectado a una fuente de voltaje Rigol que lo alimenta con 2V de forma constante, por otra parte en cada una de las terminaciones de las resistencias es conectada a través de un cable y en orden creciente de voltaje a las entradas del pmod AD5, los datos serán exportados desde la tarjeta a través del puerto de comunicación UART†† en dirección a la computadora, como se puede ilustra en las imágenes de la figura 3



**Figure 3.** Setup de lectura para modo pseudo diferencial. En la primera imagen de la derecha, se muestra el setup completo (fuente, circuito, tarjeta), la segunda es un acercamiento al circuito y la tercera un acercamiento a la conexión con el pmod.

\*\* existe una larga lista de lenguaje desarrollados para este propósito, entre los que destaca C, verilog y SystemVerilog. ya que son los que se ha utilizado a lo largo de este desarrollo

†† Universal Asynchronous Receiver-Transmitter

## 5. Implementación Digital

Siguiendo los tutoriales de Digilent antes nombrados ([link](#)) es fácil llegar a la representación del esquema de la figura 4, donde se muestran 4 bloques cada una función específica; Zynq, representa al procesador y permite realizar su configuración de funcionamiento; Reset, bloque de control que reinicia el circuito cada vez que es necesario o impuesto por el usuario; AXI interconnect bloque centrado en la interconexión y transferencia de información entre maestro y esclavo; y el PMODAD5\_v1.0 que caracteriza la interfaz con el pmod AD5 y sus conexiones.

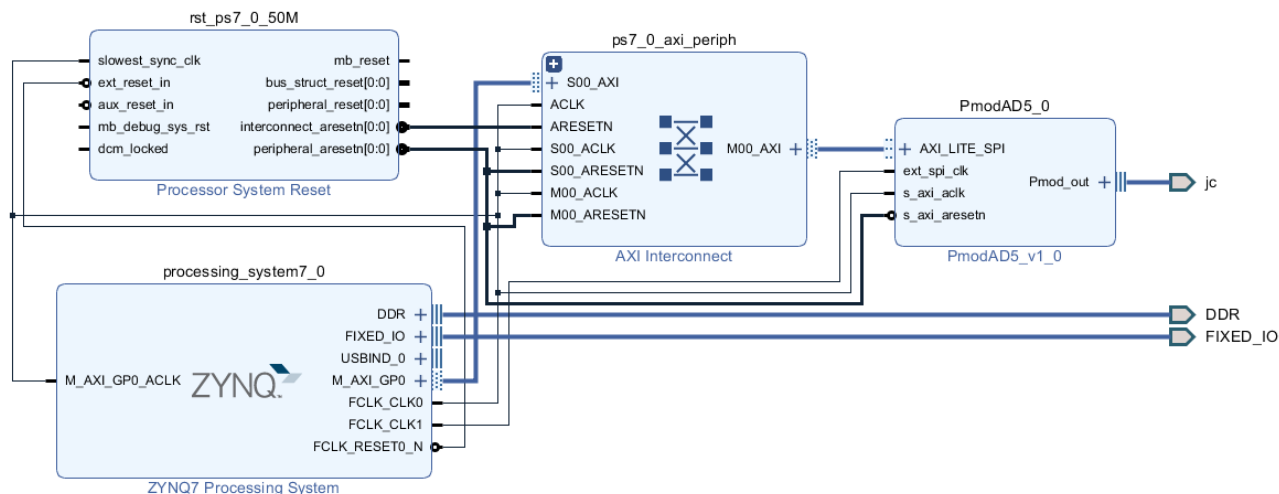


Figure 4. Captura de imagen de la configuración de hardware realizada en el software Vivado2018.2.

## 6. Códigos de programación

Siguiendo con el tutoría es fácil entender como llegamos a la implementación de un código escrito en C para la configuración del procesador(disponible a [aquí](#)), este permite la lectura de los datos desde el pmod y el envío hacia la computadora a través de algunas simples líneas de código, los cuales son leídos y representados de forma gráfica por el código python disponible [aquí](#). En particular y gracias a la existencia de la librería dedicada al pmod (disponible en este [link](#) y consta con el código de ejemplo), es que solo se plantearán las principales funciones de la librería, necesarias para tomar el control de co

[aquí](#). En particular y gracias a la existencia de la librería dedicada al pmod (disponible en este [link](#) y consta con el código de ejemplo), es que solo se plantearán las principales funciones de la librería, necesarias para tomar el control de configuración y lectura del pmod.

### 6.1. D5\_WriteConfig

Función encargada de la escritura del registro configuración del pmod AD5. Este registro consta de 24 bits cada uno asociado a una tarea específica y se encuentran perfectamente descritos en la pagina 21 del [Datasheet AD7193](#) del componente.

### 6.2. AD5\_readData

Función diseñada para enviar el comando de lectura y extraer los 24 bits de datos desde el registro data del pmod.

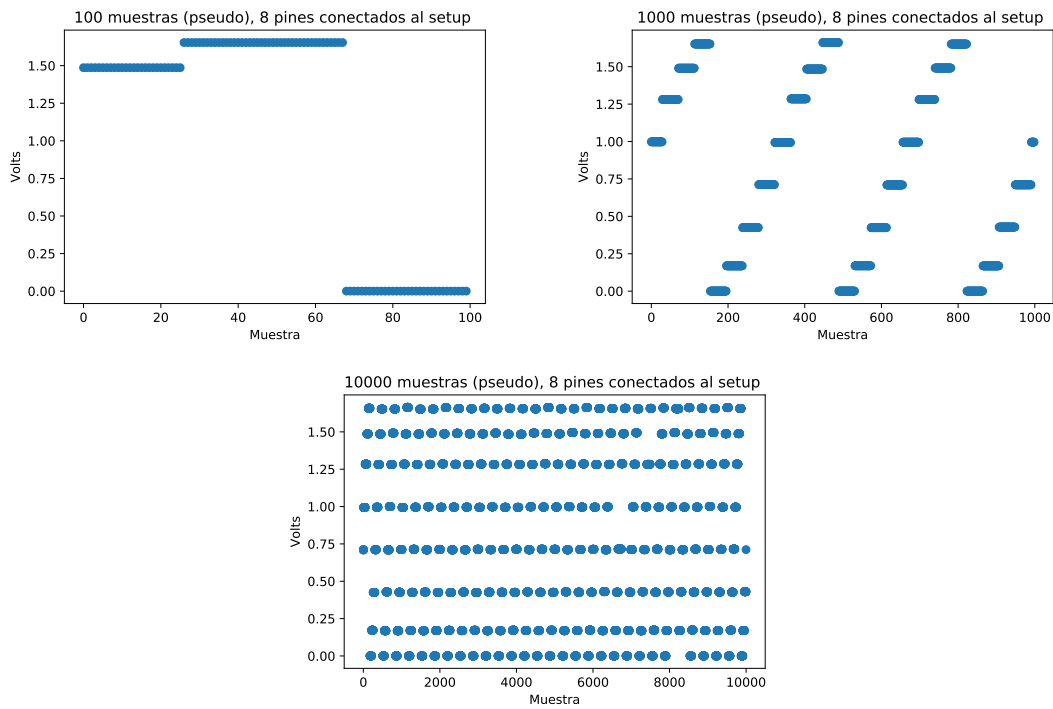
### 6.3. AD5\_DataToVoltage

Funciones diseñada para transformar el número de cuentas digitales en valores de voltaje<sup>††</sup>.

<sup>††</sup> Esta función puede ser modificada, en caso de existir variaciones en la ganancia del sistema

#### 6.4. Lectura y visualización de datos

Se implementa tanto el código en C sobre la tarjeta como el código python en la computadora (disponible en el siguiente [link](#)), la adquisición de datos se ha traducido en la manifestación de 3 gráficos diferentes, desde donde además se concluye que los datos son extraídos cada 0.0019s y el cambio de lectura de canal se produce cada 0.05s o 20 hz aproximadamente, ver figura 5.



**Figure 5.** Gráficos de lectura respecto al setup de la figura 3, las graficas muestran funcionalidad del pmod AD5.

#### 6.5. Conclusiones y trabajos futuros

Las herramientas entregadas por cada una de las diferentes fuentes presentadas en este documento son esenciales para la implementación del pmod AD5 y han demostrado ser una vía para su uso e implementación. Más aún es necesario caracterizar la respuesta de este componente con mayor precisión ya que quedan por corregir los errores de lectura, la escala de lectura e identificar correctamente cada canal de conversión.