



INAOE

**Instituto Nacional de Astrofísica, Óptica
y Electrónica.**

**Proyectos con Raspberry
PI: Sistema de adquisición
de datos para Estación
Meteorológica y Monitor
de temperaturas
criogénicas**

por

Dr. Daniel Ferrusca R.

Luis Alejandro Díaz León

Ing. Jesús Contreras R.

REPORTE TÉCNICO

©INAOE 2015

Derechos Reservados

El autor otorga al INAOE el permiso de reproducir y distribuir copias de este reporte técnico en su totalidad o en partes mencionando la fuente.



ÍNDICE GENERAL

1. OBJETIVO GENERALES	2
2. GENERALIDADES RASPBERRY	2
Empezando con la <i>Raspberry Pi</i>	3
3. Descripción de los proyectos	4
4. Estación Meteorológica.....	5
Monitor de Viento Young.....	7
Weather Logging System.....	8
Diseño del Circuito Impreso	9
Funcionamiento del sistema	10
Conclusiones	12
5. Módulo electrónico para lectura digital de Sensor Criogénico.....	13
Termómetros resistivos.....	13
Puente de Wheatstone	15
Diseño del puente utilizado para el módulo	16
Amplificador Operacional Diferencial	18
Diseño del circuito impreso.....	20
Funcionamiento del sistema	21
Conclusiones	26
6. CONCLUSIONES GENERALES	27
7. REFERENCIAS.....	27
Apendice A	28
Programa para el sistema de adquisición de datos para Estación Meteorológica	28
Apendice B	35
Programa para el sistema de adquisición de datos para sensor criogénico	35

1. OBJETIVO GENERALES

- Realizar dos proyectos de aplicación usando la *Raspberry Pi*.
- Desarrollar una estación meteorológica portátil, de bajo costo utilizando la *Raspberry Pi*, para monitorear los cambios climáticos del sitio de Gran Telescopio Milimétrico (GTM).
- Diseñar un módulo electrónico de bajo costo para un sensor criogénico, para obtener mediciones de temperaturas de $-200\text{ }^{\circ}\text{C}$.

2. GENERALIDADES RASPBERRY

Raspberry Pi es una computadora de placa reducida de bajo costo desarrollada en Reino Unido por la *Fundación Raspberry Pi*, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas. (Fig. 1)

El diseño incluye un System-on-a-chip Broadcom BCM2835, que contiene un procesador central (CPU) ARM1176JZF-S a 700 MHz, un procesador gráfico (GPU) VideoCore IV, y 512 MB de memoria RAM. Usa mayoritariamente sistemas operativos basados en el núcleo Linux. Raspbian, una distribución derivada de Debian que está optimizada para el hardware de Raspberry Pi.

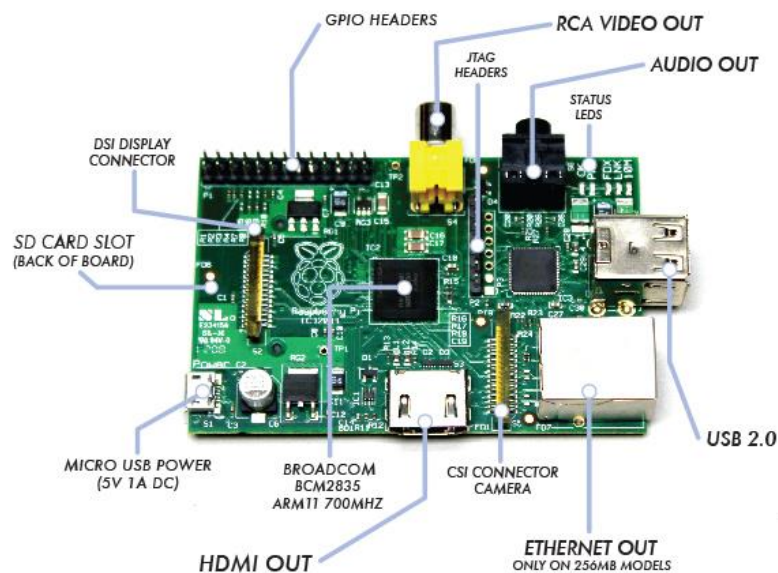


IMAGEN 1 – TARJETA RASPBERRY PI UTILIZADA EN LOS PROYECTOS DESCRITOS.

Usando esta tarjeta se pueden desarrollar distintos proyectos y diversas aplicaciones gracias a su potencia y versatilidad. Desde un simple circuito de lectura y escritura de datos, un set de vigilancia hasta un sistema de automatización y robótica. Mayor información sobre esta tarjeta pueda encontrarse en la sección de referencias o en internet.

EMPEZANDO CON LA *RASPBERRY PI*

Al trabajar por primera vez con la *Raspberry Pi* la configuración correcta es parte fundamental para su óptimo funcionamiento.

1. El primer paso es contar con una tarjeta SD de por lo menos 4GB para grabar la imagen del sistema operativo, el *Raspbian*. El ambiente del sistema operativo está basado en Linux , trabajando de la misma forma y con los mismos comandos.
2. Lo siguiente es conectar en los puertos USB un mouse y un teclado para trabajar en el modo gráfico. Seguido de conectar un monitor vía puerto HDMI para visualización.
3. Para actualizar el firmware se necesita estar conectado a internet, para ello se conecta directamente al modem por cable de Ethernet.
4. En el modo gráfico se puede trabajar de igual manera que una computadora normal, con las operaciones básicas disponibles como editor de texto, navegador web, audio, video y los ambientes de programación como *Scratch* y *Python* por default.
5. Para correr los programas hechos en *Python*, primero se escriben en el editor de texto y se les asigna la extensión *.py* y ejecutandose en la Terminal LX.
6. Para trabajar con *Python* y controlar la *Raspberry* se necesitan de los paquetes y bibliotecas correspondientes y así utilizar todas las funciones disponibles (Los proyectos descritos en este reporte fueron programados en lenguaje *Python*).

3. DESCRIPCIÓN DE LOS PROYECTOS

En la realización de los proyectos aquí descritos, se aplicaron las funciones de lectura y escritura de datos usando los puertos GPIO de la tarjeta, un ADC externo para la interpretación de los valores analógicos y un circuito con amplificadores operacionales para el acondicionamiento de señal.

Para el proyecto de adquisición de datos de la estación meteorológica se utilizó un anemómetro para las lecturas de velocidad y dirección del viento, así como también, un data logger con sensores de humedad y temperatura. Una vez finalizada, fue llevada al sitio del GTM para su montaje.

El diseño para el módulo de adquisición de datos del sensor criogénico se desarrolló aplicando teoría de circuitos y amplificación para obtener valores legibles para el ADC y poderlos interpretar en niveles de temperatura. El sensor criogénico al ser del tipo resistivo, necesita de un arreglo de otras resistencias para calcular la diferencia de potencial para determinar su magnitud resistiva que será comparada con el valor de temperatura correspondiente en una tabla proporcionada por el fabricante.

4. ESTACIÓN METEOROLÓGICA

Una estación meteorológica consiste en una unidad con diferentes sensores para monitorear las condiciones climatológicas de un sitio ya sea de forma remota o directa

El programa diseñado para este proyecto utiliza la librería *WiringPi* disponible para *Python* con ciertos módulos de ADC, que ya vienen configurados y listos para leer datos en cada canal y para comunicarse con la *Raspberry* vía SPI(*Serial Protocol Interface*). Esta comunicación consiste en los siguientes pines para llevar a cabo la comunicación entre ambos dispositivos.

- MISO (Master In Slave Out)
- MOSI (Master Out Slave In)
- SCK (Serial Clock)
- CS (Chip Select)

Transferencia SPI

Una transferencia SPI es iniciada por el Master poniendo en BAJO CS. La línea de CS se mantiene en ALTO durante el estado de reposo. Ahora el Master puede escribir en el bus de 8 bits (1 byte). Una cosa más importante a tener en cuenta sobre SPI es que por cada byte Maestro escribe a Slave, Master recibe un byte a cambio. Así que la única operación posible es el intercambio de datos.

MCP3204 12 bit SPI ADC Chip

El ADC utilizado fue el *MCP3204*, circuito integrado de 4 canales, comunicación SPI y una resolución de 12bits. La conversión analógica-digital consiste en la transcripción de señales analógicas en señales digitales, con el propósito de facilitar su procesamiento y hacer la señal resultante más inmune al ruido y otras interferencias a las que son más sensibles las señales analógicas.

Exactitud y Resolución de un Convertidor ADC

Se tiene que determinar con que exactitud será la conversión entre la señal analógica y la digital, para lo cual se define la resolución que ésta tendrá.

Primero se define el número máximo de bits de salida. Este dato permite determinar el número máximo de combinaciones en la salida digital. Este número máximo está dado por: 2^n donde n es el número de bits.

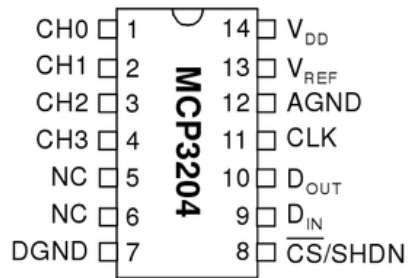


IMAGEN 2 – CIRCUITO INTEGRADO UTILIZADO COMO CONVERTIDOR A/D.

También la resolución se entiende como el voltaje necesario (señal analógica) para lograr que en la salida (señal digital) haya un cambio del bit menos significativo (LSB).

LSB significa: Least Significant Bit

Para hallar la resolución se utiliza la fórmula

$$Resolución = \frac{V_{ref}}{[2^n - 1]}$$

Donde:

n = es el número de bits que tiene el Convertidor Analógico Digital

V_{ref} = es el voltaje que hay que poner a la entrada del convertidor ADC, para obtener una conversión máxima (todas las salidas serán iguales a "1").

El anemómetro utilizado manda una señal de 0-5V, por lo tanto nuestro voltaje de referencia será de 5V. La resolución es de 12 bits, esto es, 4096 pasos.

$$Resolución = \frac{5}{4095}$$

Con esta ecuación se determina que cada paso incrementará .001222V de 0-4095(12 bits) hasta llegar a 5V.

Después de obtener la medición correcta, los valores son comparados con la escala determinada por el fabricante del anemómetro, para desplegar los valores de velocidad y dirección del viento en el programa.

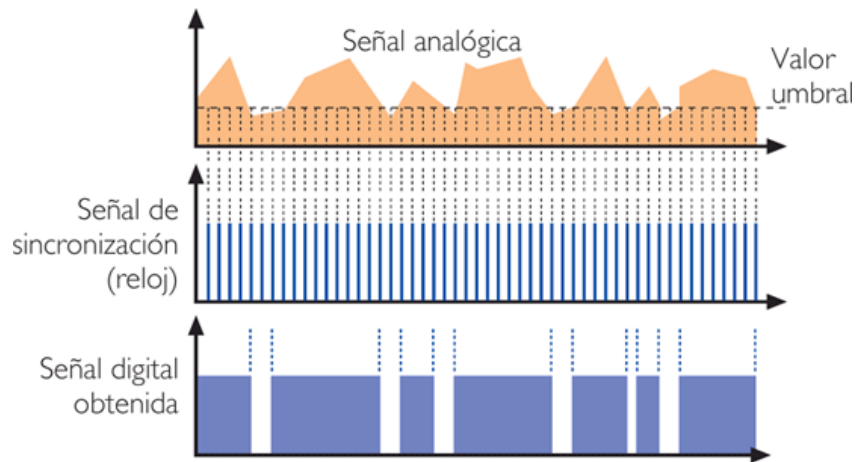


IMAGEN 3 – ESQUEMA DE CONVERSION ANALOGICO DIGITAL CON SINCRONIZACION DE RELOJ

Para determinar de manera más exacta el valor de ambas variables, se calculó el promedio de 10 lecturas y desplegándose cada 10 segundos. Todas estas lecturas desplegadas son las que deben registrarse (el programa cuenta con fecha y hora para el registro de estas variables).

MONITOR DE VIENTO YOUNG

El monitor utilizado para obtener las variables del viento antes mencionadas fue el *Young: modelo 05103*; un sensor de viento de alto rendimiento. El sensor de velocidad del viento es una hélice helicoidal de cuatro palas. La rotación de la hélice produce una señal de tensión de onda sinusoidal AC. El sensor de dirección del viento es una paleta, robusto pero ligero. El ángulo es detectado por un potenciómetro de precisión. Este modelo ofrece salidas de 0-5V DC en sus dos canales de salida.

La variable de velocidad incrementa .05V cada m/s, siendo 100m/s la velocidad máxima que puede detectar (velocidad que en condiciones normales no se alcanza) hasta llegar a los 5V.

$$Vel = \frac{Volts}{.05}$$



IMAGEN 4 – ANEMOMETRO YOUNG UTILIZADO EN PRUEBAS DE LABORATORIO Y EN LAS INSTALACIONES DEL GRAN TELESCOPIO MILIMETRICO PARA LECTURA A/D CON RASPBERRY PI

El ángulo de dirección de viento incrementa .0139V cada grado llegando como máximo a 355° debido al ancho del cuerpo del monitor que incluye esos 5° faltantes.

$$Dir = \frac{Volts}{.0139}$$

WEATHER LOGGING SYSTEM

Para las variables de temperatura, presión y humedad, se utilizó el Data *Logger WLS-8000* de Texas Weather Instruments con los sensores correspondientes de cada variable. Estas variables ya están calculadas y calibradas, por lo que solo se necesita extraerlas y desplegarlas.

La comunicación es necesaria para extraer estas variables del Data Logger es vía Serial, por lo que se debe habilitar la recepción y transmisión de datos en el programa. Se necesita mandar un comando específico al instrumento para recibir las variables que se necesitan (en este caso fue "Q" en ASCII). Las variables son enviadas a la *Raspberry* usando un adaptador Serial-USB y son desplegadas en cadena de caracteres en un orden predeterminado la cual puede ser dividida para solo tomar las variables necesarias (temperatura, humedad y presión). Estas se añaden a las otras variables del anemómetro para desplegarse todas al mismo tiempo en una nueva cadena de caracteres que será registrada en un archivo de texto con la fecha y hora correspondiente. Dicho archivo es creado bajo el nombre de la fecha de cada día, generándose uno nuevo de forma automática al pasar 24hrs.

Si se requiere monitorear las variables en tiempo real se puede crear una base de datos con la misma cadena de caracteres y ser visualizada en una gráfica o en una página web.



IMAGEN 5 – DATA LOGGER TEXAS WEATHER INSTRUMENTS UTILIZADO EN PRUEBAS DE LABORATORIO Y EN LAS INSTALACIONES DEL GRAN TELESCOPIO MILIMETRICO PARA LECTURA A/D CON RASPBERRY PI

DISEÑO DEL CIRCUITO IMPRESO

El diseño final de este proyecto consistió en la fabricación de un circuito impreso del ADC para ser incorporado a la Raspberry como un módulo externo. De esta manera se podrán encapsular como un solo dispositivo facilitando su instalación y conexión con los dispositivos de monitoreo.

El módulo externo se conecta directamente a la tarjeta con los pines GPIO, quedando ensamblada encima de ésta optimizando espacio y dejando libre los canales del ADC para una fácil conexión con los sensores. Se diseñó la placa a dos capas para reducir espacio lo más posible y pudiera caber dentro del encapsulado.

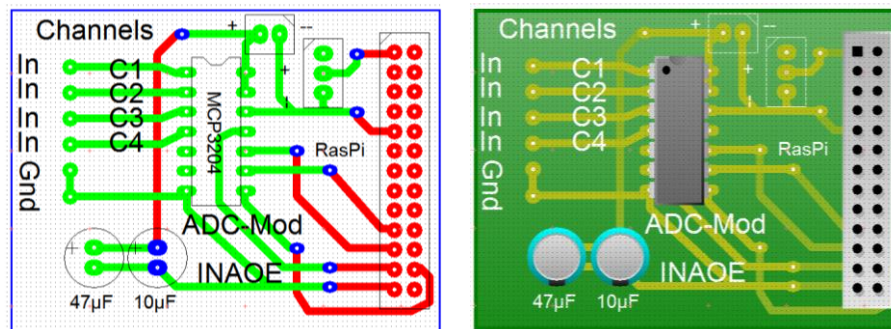


IMAGEN 6 – DISEÑO DEL CIRCUITO IMPRESO DEL ADC. LA TARJETA SE CONECTA DIRECTA A LA TARJETA MADRE DE LA RASPBERRY PI.

FUNCIONAMIENTO DEL SISTEMA

Teniendo el circuito impreso terminado, se conecta a la *Raspberry* para comenzar las lecturas. El programa ya está configurado para ejecutarse al encender la tarjeta, por lo que no hay nada que ingresar para iniciar el programa. Se debe tener conectados los dispositivos y sensores antes de encender la tarjeta para evitar lecturas basura.

Las lecturas realizadas son extraídas del archivo de texto para poder crear una gráfica del comportamiento de las variables durante el tiempo que esté operando la estación. La gráfica es desplegada en la página Web del telescopio para ser consultada en tiempo real.

El proyecto fue llevado al sitio del Gran Telescopio Milimétrico(GTM) para probar su desempeño y realizar mediciones de las condiciones del viento durante 3 semanas.

La meta de este proyecto fue que la estación fuera eficiente dentro de su bajo costo, y que además, pudiera soportar las condiciones extremas que hay en el sitio.

La estación logró operar correctamente de forma continua durante el tiempo de prueba con lecturas precisas y concordantes con las mediciones de otros anemómetros en el sitio. Esto permitió determinar su buena eficiencia y confiabilidad para ser implementada de forma permanente en un futuro.

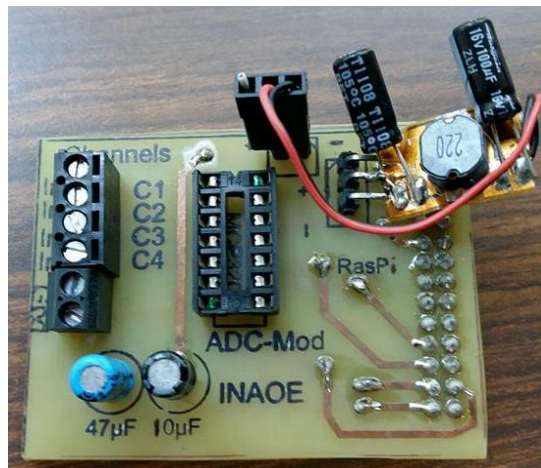


IMAGEN 7 – MODULO ADC ENSAMBLADO. SE MUESTRA CIRCUITO BOOSTER NIVELACION DE VOLTAJES DE REFERENCIA UTILIZADOS EN CONJUNTO CON LA RASPBERRY PI.



IMAGEN 8 – ENCAPSULADO DE LA RASPBERRY PI



IMAGEN 9 – MÓDULO ADC INSERTADO A RASPBERRY PI



IMAGEN 10 – ENCAPSULADO COMPLETO DE TARJETA RASPBERRY PI, MODULO ADC Y BOOSTER CON CONECTIVIDAD A PC

CONCLUSIONES

Lo complicado de este proyecto fue alcanzar la máxima precisión posible al tratarse de un diseño portátil, ya que se utilizó una sola fuente de alimentación para toda la estación. Esto, con el objetivo de simplificar el diseño y ser más compacto el encapsulado como también tener la posibilidad de operar con baterías.

La *Raspberry* trabaja con una fuente de 5V a 1A mínimo para un funcionamiento óptimo, de igual forma el ADC para su máxima velocidad de muestreo. Éste necesita también 5V en el pin de voltaje de referencia para cuantificar en la escala completa y tener lecturas precisas. Debido a la caída de voltaje en dicho pin, se recurrió a un "*Booster de 5V*" para elevar el voltaje y mantenerlo en 5V. Con esto se soluciona el problema de la escala completa de precisión, pero el *Booster* exige cierto nivel de corriente que provoca el calentamiento de la tarjeta, cuestión que puede solucionarse a futuro con la sustitución de la fuente de alimentación por una de mayor corriente.

Otra propuesta para realizar el proyecto con un excelente desempeño evitando los problemas de energía, sería el contar con una segunda fuente de alimentación independiente para el ADC.

5. MÓDULO ELECTRÓNICO PARA LECTURA DIGITAL DE SENSOR CRIOGÉNICO

El siguiente proyecto consistió en fabricar un módulo para lectura digital del sensor PT100 de *Lake Shore*, utilizado para medir temperatura. Para ello, se necesitó diseñar un circuito capaz de adecuar y amplificar la señal recibida del sensor para ser analizada y desplegada.

TERMÓMETROS RESISTIVOS

Los termómetros de resistencia o detectores de temperatura de resistencia también llamados (RTD), son sensores utilizados para medir la temperatura mediante la correlación de la resistencia del elemento RTD con la temperatura. La mayoría de estos sensores consisten en una longitud de alambre en espiral fina envuelta alrededor de un núcleo de cerámica o de vidrio. El elemento suele ser bastante frágil, lo que a menudo se coloca dentro de una sonda de envainado para protegerlo. Un RTD está hecho de un material puro, típicamente platino, níquel o cobre. El material tiene un cambio predecible en la resistencia conforme el cambio de temperatura y es este cambio predecible el que se utiliza para determinar la temperatura.

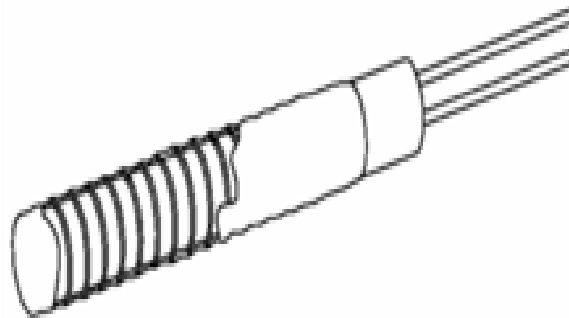


IMAGEN 11 – SENSOR RESISTIVO DE TEMPERATURA RTD, PT100

El sensor PT100 puede detectar temperaturas bajas en un rango de 0 hasta -200 °C. La resistencia de referencia del PT100 es de 100 Ohms a 0°C, la cual es la temperatura de calibración.

El platino es el mejor metal para RTDs porque sigue una relación resistencia - temperatura casi lineal y sigue la relación R vs T de una manera altamente repetible en un amplio rango de temperaturas. Las propiedades únicas de platino lo convierten en el material de elección para los estándares de temperatura en el rango de -272.5 ° C a 961,78 ° C , y se utiliza en los sensores que definen la Norma Internacional de temperatura, ITS - 90.

La característica significativa de metales usados como elementos resistivos es la aproximación lineal de la resistencia frente a la relación de temperatura entre 0 y 100 ° C .

El sensor cuenta con una tabla dada por el fabricante con los valores exactos de temperatura según la resistencia la cual sirve de referencia, sin embargo, para calcular de manera matemática la gráfica de temperatura se utiliza el Alpha de 0.00385 especificada por las normas reconocidas para RTD industriales IEC 60751 y ASTM E - 1137.

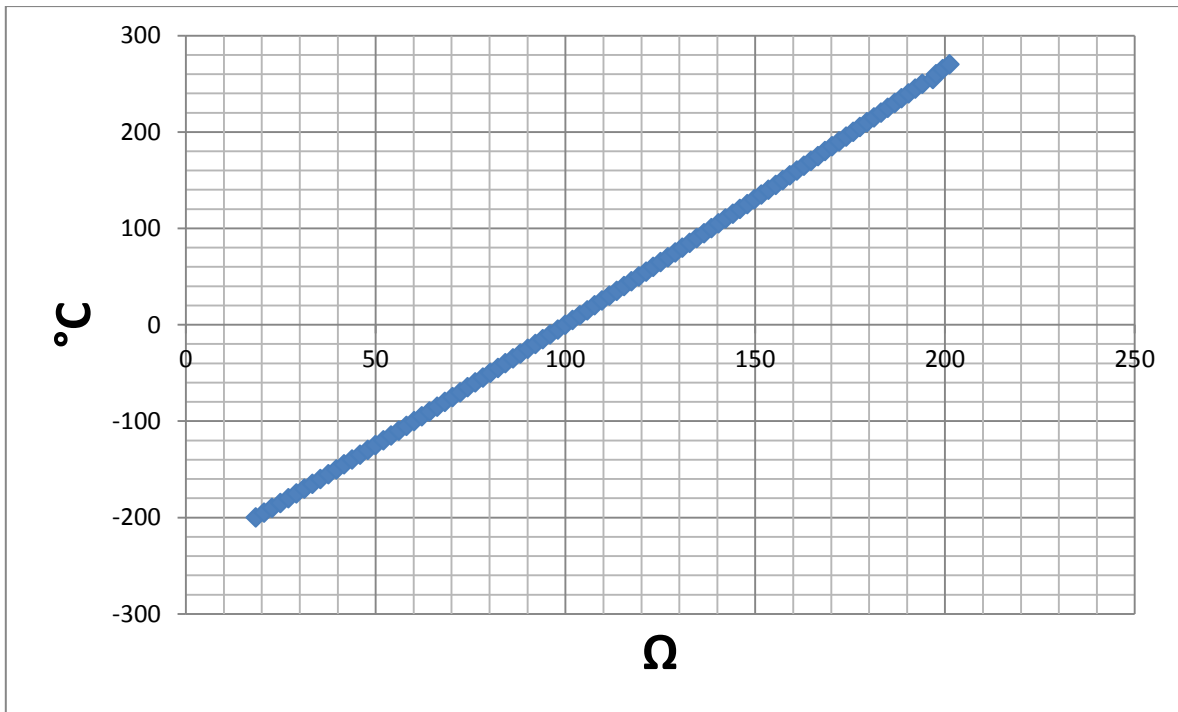


IMAGEN 12 – CURVA R-T DEL SENSOR PT-103 DADA POR EL FABRICANTE

La relación de Resistencia con Temperatura se calcula con la expresión generada de la aproximación lineal con Alpha.

$$RTD = R0(1 + \alpha\Delta T)$$

Despejando la temperatura:

$$T = \frac{1}{\alpha} * \left[\frac{RTD}{R0} - 1 \right]$$

Donde:

T es la temperatura en °C

$\alpha = .00385$

RTD la resistencia actual del sensor en Ohms

R0 la resistencia de referencia a 0°C(100 Ohms)

Una de las complicaciones para el diseño del módulo, fue el circuito de acondicionamiento de señal, debido a que el sensor al disminuir su temperatura, la resistencia llega a ser muy baja y genera señales muy débiles dificultando su medición a causa del ruido. Deben respetarse ciertas especificaciones del sensor para evitar errores de medición o dañarlo. Estas son principalmente de corriente y potencia disipada. La corriente máxima que debe pasar por el sensor es de .05mA y 10uW de potencia. Al ser estos valores demasiado bajos se requiere de un arreglo especial para dividir el voltaje y no exigirle corriente al sensor directamente. Esto se logra con el arreglo de resistencias de *Puente de Wheatstone*.

PUENTE DE WHEATSTONE

El puente de Wheatstone ayuda a calcular las medidas de resistencias desconocidas mediante el equilibrio de los brazos que usa el puente. Estos están contruidos por cuatro resistencias que forman un circuito cerrado, siendo una de ellas la resistencia de bajas medidas.

Para encontrar el valor de la resistencia del sensor se necesita primero saber la diferencia de voltaje que existe entre los brazos del puente. Dicha diferencia de voltaje es el resultado de la división de voltaje de cada brazo del puente, lo que significa, que depende directamente del valor de las resistencias. Con ese voltaje se puede encontrar el valor que tendría la resistencia del sensor y por consiguiente, la temperatura.

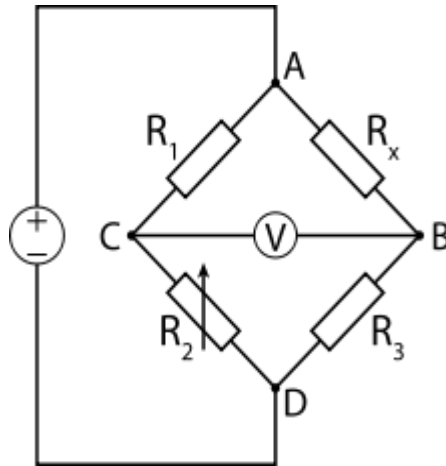


IMAGEN 13 – ESQUEMA GENERAL DE PUENTE DE WHEATSTONE

El voltaje entre los brazos del puente se calcula de la siguiente manera:

$$V_m = V_3 - V_{RTD}$$

Entonces sí:

$$V_3 = V_{in} \frac{R_4}{R_4 + R_2} \quad \text{y} \quad V_{rtd} = V_{in} \frac{RTD}{RTD + R_1}$$

$$V_m = \frac{R_4}{R_4 + R_2} - \frac{RTD}{RTD + R_1}$$

DISEÑO DEL PUENTE UTILIZADO PARA EL MÓDULO

Donde:

$$R_1 = 1K$$

$$R_2 = 150\Omega$$

$$R_3 = 260\Omega \text{ (resistencia limitadora del sensor)}$$

RTD = (valor de la resistencia del sensor)

R4 = 1K

Vin = 210mV (valor que satisface las especificaciones de corriente y potencia)

Vm = Valor de la diferencia de voltaje entre los brazos del puente

Despejando RTD y sustituyendo los valores, la ecuación final queda:

$$RTD = \frac{182.60113 - 1000Vm}{vm+.027298869} - 260$$

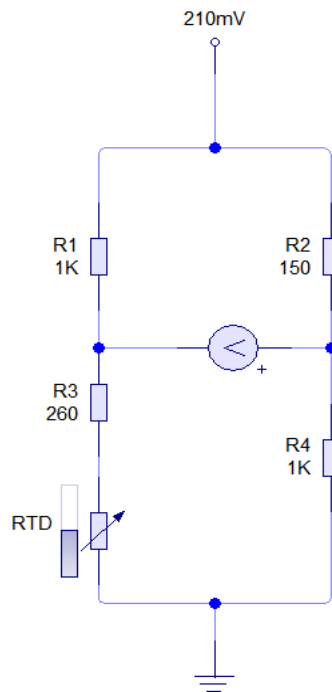


IMAGEN 14 – DISEÑO FINAL DEL PUENTE DE WHEATSTONE UTILIZADO EN EL PROYECTO

Con esta ecuación se obtiene el valor exacto de la resistencia del sensor para cualquier voltaje de referencia.

Sin embargo, la señal proveniente al estar dividida en dos voltajes diferentes, no puede interpretarse directamente. El ADC reconoce una diferencia de potencial con respecto a tierra. Ambas señales deben ser interpretadas como una sola señal y tener tierra como nodo de referencia. Esto se logra mediante un amplificador diferencial.

AMPLIFICADOR OPERACIONAL DIFERENCIAL

El amplificador operacional es un dispositivo activo con diversas aplicaciones dentro de la electrónica analógica, su salida consiste en la diferencia entre sus entradas multiplicada por un factor de ganancia G . Existen muchas configuraciones diferentes según la aplicación y para este proyecto se implementó la configuración diferencial. Esta configuración, nos permite encontrar la diferencia entre dos señales independientes. El arreglo de las cuatro resistencias determina la ganancia de la salida y el equilibrio entre ambas entradas, que deben tener la misma impedancia de entrada.

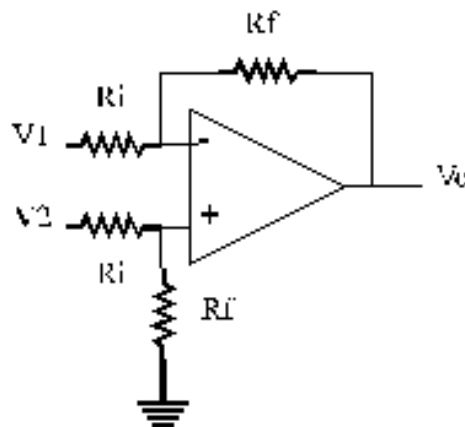


IMAGEN 15 – ESQUEMA GENERAL DE AMPLIFICADOR DIFERENCIAL

En este caso como solo se necesitaba obtener la diferencia entre dos señales por lo que se configuró con una ganancia unitaria; $G=1$. Por lo tanto, las cuatro resistencias deben ser del mismo valor.

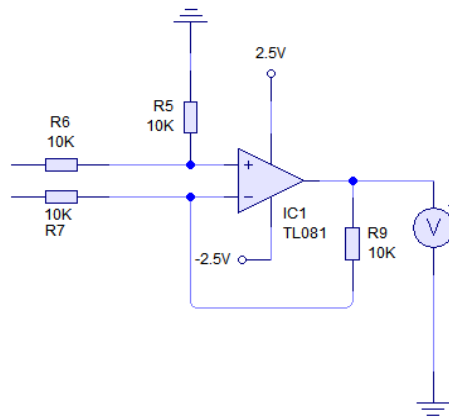


IMAGEN 16 – DISEÑO DEL AMPLIFICADOR DIFERENCIAL UTILIZADO EN EL PROYECTO

La salida de cada divisor de voltaje del *Puente de Wheatstone* se conecta a una de las entradas del amplificador para obtener la diferencia resultante en la salida. Esta nueva señal al estar en referencia con tierra, ya puede ser interpretada por el ADC.

En el circuito completo, se adicionó un seguidor de voltaje para cada salida del puente para evitar pérdida de señal debido a las resistencias al conectarse directamente y tener lecturas erróneas. Con esto el voltaje se mantendrá fijo para las entradas del amplificador.

Todos los amplificadores necesitan de una fuente simétrica para operar con el rango máximo de amplificación(+/- Vcc). Pero dado a la complejidad que esto conlleva y para simplificar el circuito, se utilizó en Amplificador Operacional LM358. Este particular integrado es ideal para este tipo de aplicaciones portátiles y simples ya que es el único que puede ser operado con una sola fuente de alimentación. Lo que permitirá proporcionar el rango máximo sin problemas y con un diseño simple. Teniendo en cuenta que debe ser alimentado por un voltaje suficiente para lograr abarcar el rango requerido.

El diseño final cuenta con un regulador de voltaje y un seguidor para proporcionar un voltaje fijo al puente y evitar pérdidas así como también cumplir con las especificaciones. Los rangos del voltaje de diferencia dependerán del cambio de la resistencia de sensor al medir temperatura.

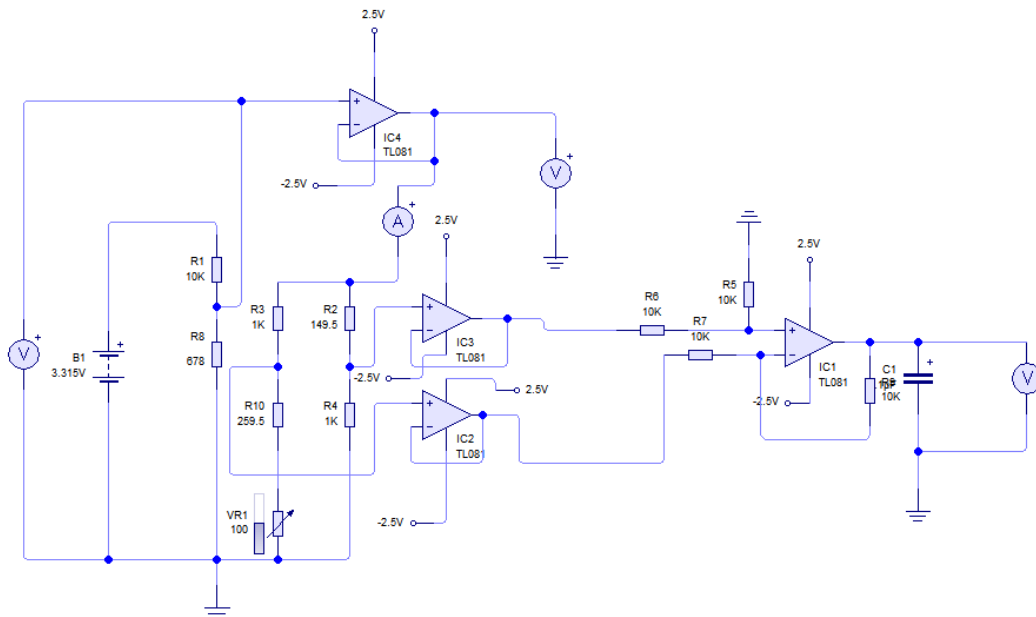


IMAGEN 17 – DISEÑO FINAL DEL AMPLIFICADOR DIFERENCIAL Y PUENTE DE WHEATSTONE

DISEÑO DEL CIRCUITO IMPRESO

El diseño final de este proyecto consistió en la fabricación del módulo para el sensor RTD y nuevamente uno del ADC para ser incorporados ambos a la Raspberry como módulos externos. De esta manera se podrán encapsular como un solo dispositivo facilitando su instalación y conexión.

El módulo del ADC se conecta directamente a la tarjeta con los pines GPIO, quedando ensamblada encima de ésta optimizando espacio y dejando libre los canales del ADC para una fácil conexión para otras señales. Este nuevo diseño se modificó de tal manera que tuviera una terminal especial para ensamblar el módulo del RTD encima. Se diseñaron ambas placas a dos capas para reducir espacio lo más posible y pudieran caber dentro del encapsulado. Finalmente se adaptó un conector DB25 hembra a la carcasa con una conexión directa con el módulo y sea la interface con el sensor por el conector macho.

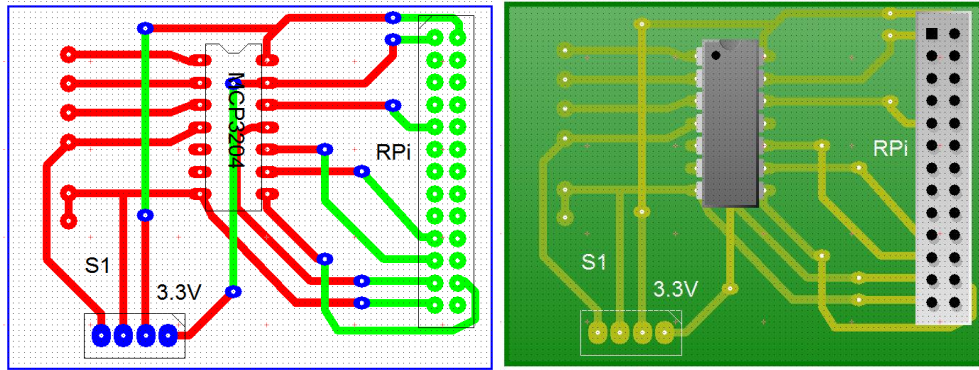


IMAGEN 18 – DISEÑO DEL PCB DEL ADC

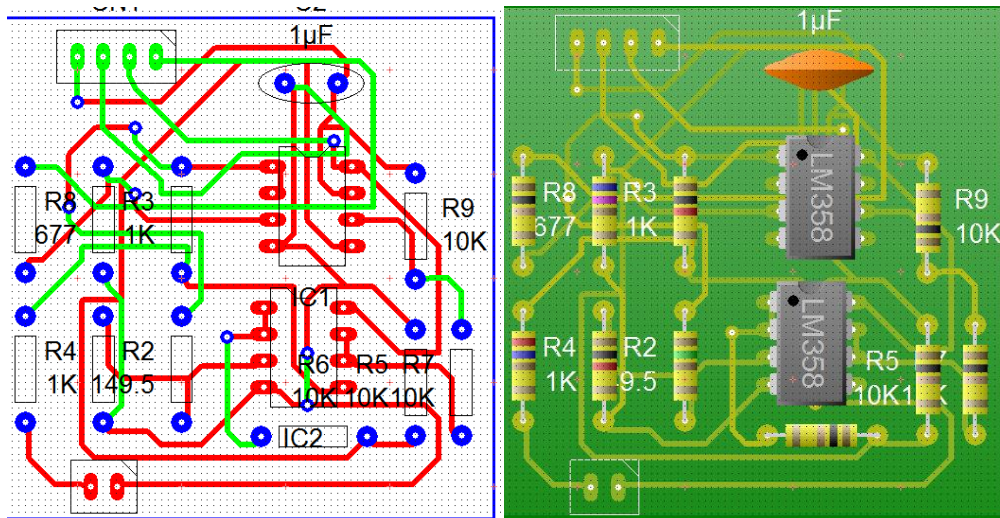


IMAGEN 19 – DISEÑO DEL PCB DEL MÓDULO ELECTRÓNICO DEL SENSOR

FUNCIONAMIENTO DEL SISTEMA

Teniendo ambos circuitos impresos terminados, se conectan a la *Raspberry* para comenzar las lecturas del sensor. El programa ya está configurado para ejecutarse al encender la tarjeta, por lo que no hay nada que ingresar para iniciar el programa. Se debe tener conectados ambos módulos y el sensor antes de encender la tarjeta para evitar lecturas basura.

Una vez encendida la tarjeta se coloca el nitrógeno líquido en un recipiente y se sumerge el sensor. De esta manera se notará como la temperatura baja abruptamente y se mantiene constante alrededor de -212°C (60K) con un margen de error de $(\pm 5 \text{ grados})$. Al

consumirse el nitrógeno, se notará como irá aumentando la temperatura gradualmente hasta llegar a la temperatura ambiente. Existirán variaciones en las mediciones durante esta transición debido a la velocidad con la que equilibra la temperatura en el sensor.

Las lecturas obtenidas fueron comparadas con los valores del fabricante y con los valores de la aproximación lineal ideal, arrojando los siguientes resultados:

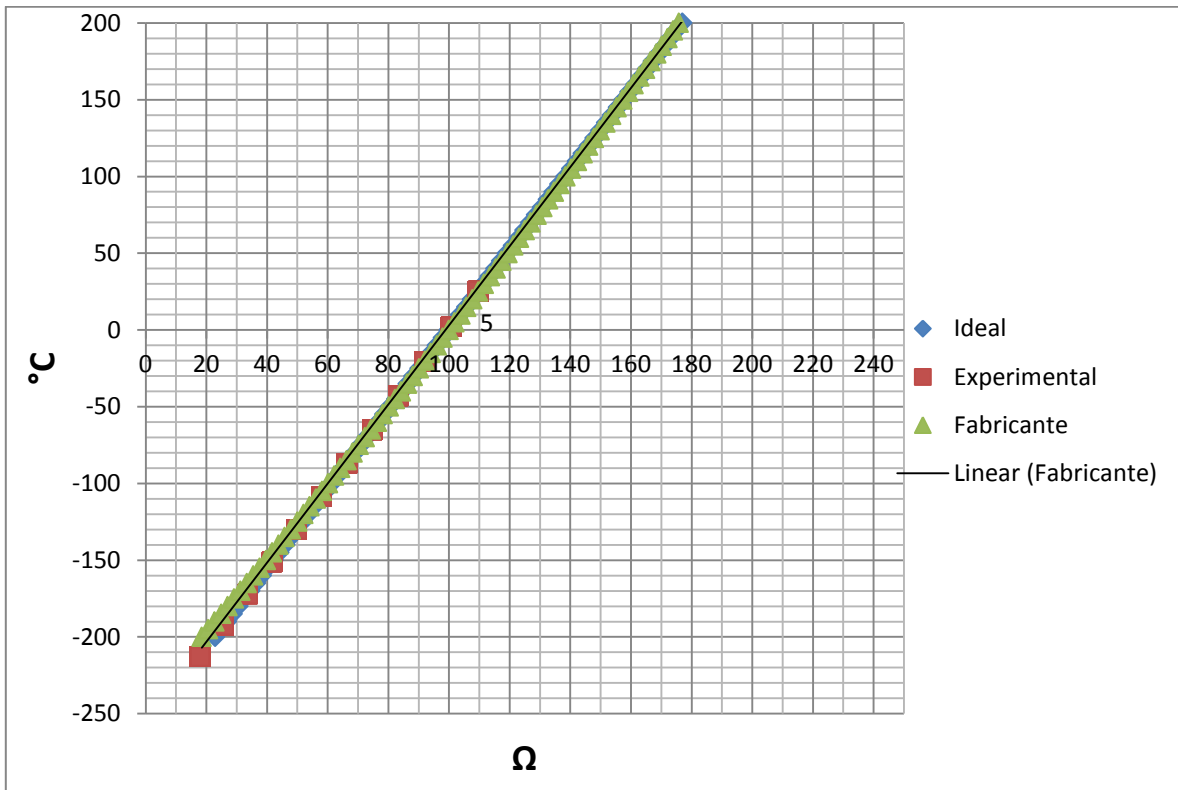


IMAGEN 20 – TABLA COMPARATIVA ENTRE GRÁFICAS DE FABRICANTE-IDEAL-EXPERIMENTAL

Se puede observar que la diferencia entre ellas es realmente muy poca, lo cual resulta ser un excelente resultado para haber sido diseñado e implementado con material de bajo costo. Considerando que existe el aparato diseñado por el fabricante que realiza esta tarea y es el más preciso pero de muy alto costo.

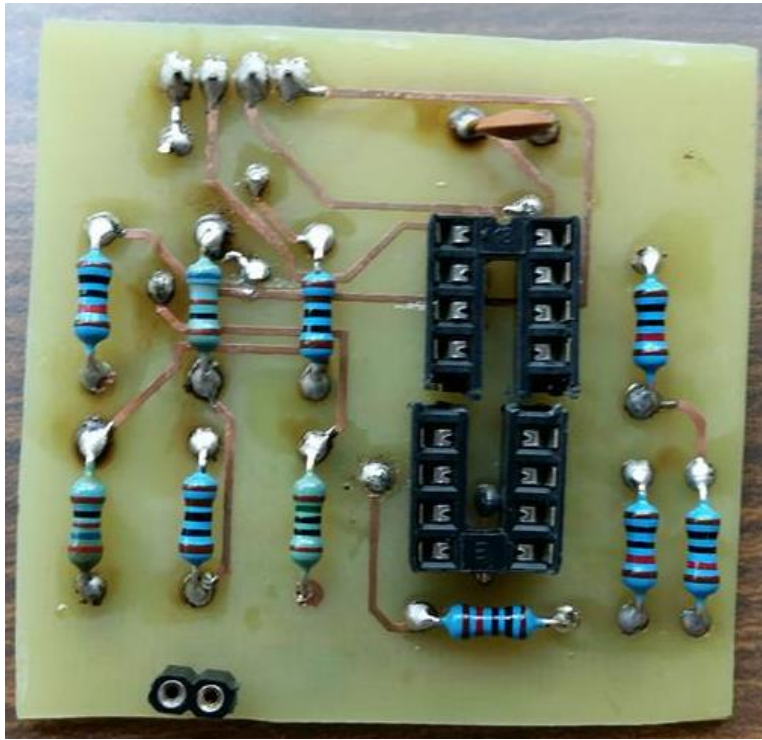


IMAGEN 21 – MÓDULO CON AMPLIFICADORES OPERACIONALES

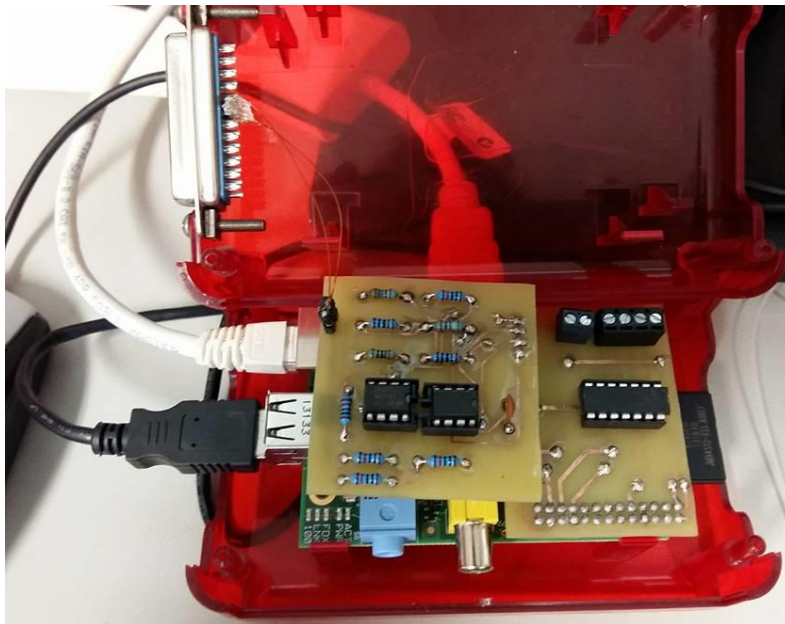


IMAGEN 22 – ENSAMBLE ENTRE LOS MÓDULOS ELECTRONICOS

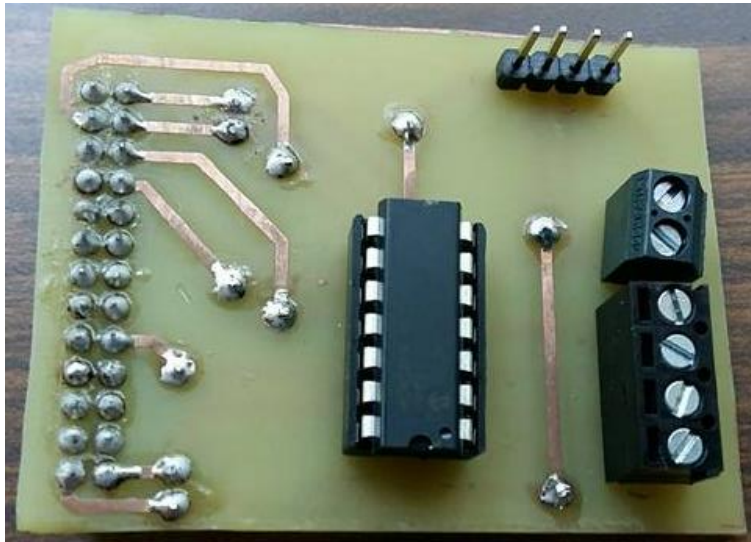


IMAGEN 23 – MÓDULO ADC PARA EL

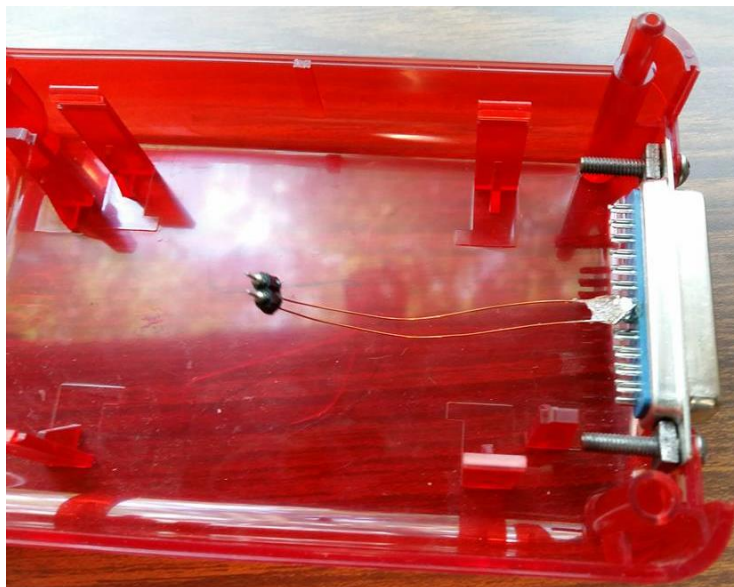


IMAGEN 24 – CARCASA CON CONECTOR DB25 HEMBRA PARA CONEXIÓN A SENSOR CRIOGÉNICO



IMAGEN 25 – CONEXIÓN FINAL DE RASPBERRY PI Y TARJETA A/D Y AMPLIFICADORA CON CONEXIÓN A PC

En la siguiente tabla se muestra el comportamiento del sensor al hacer las medidas experimentales enfriándolo y dejando que se evapore el nitrógeno hasta llegar a temperatura ambiente.

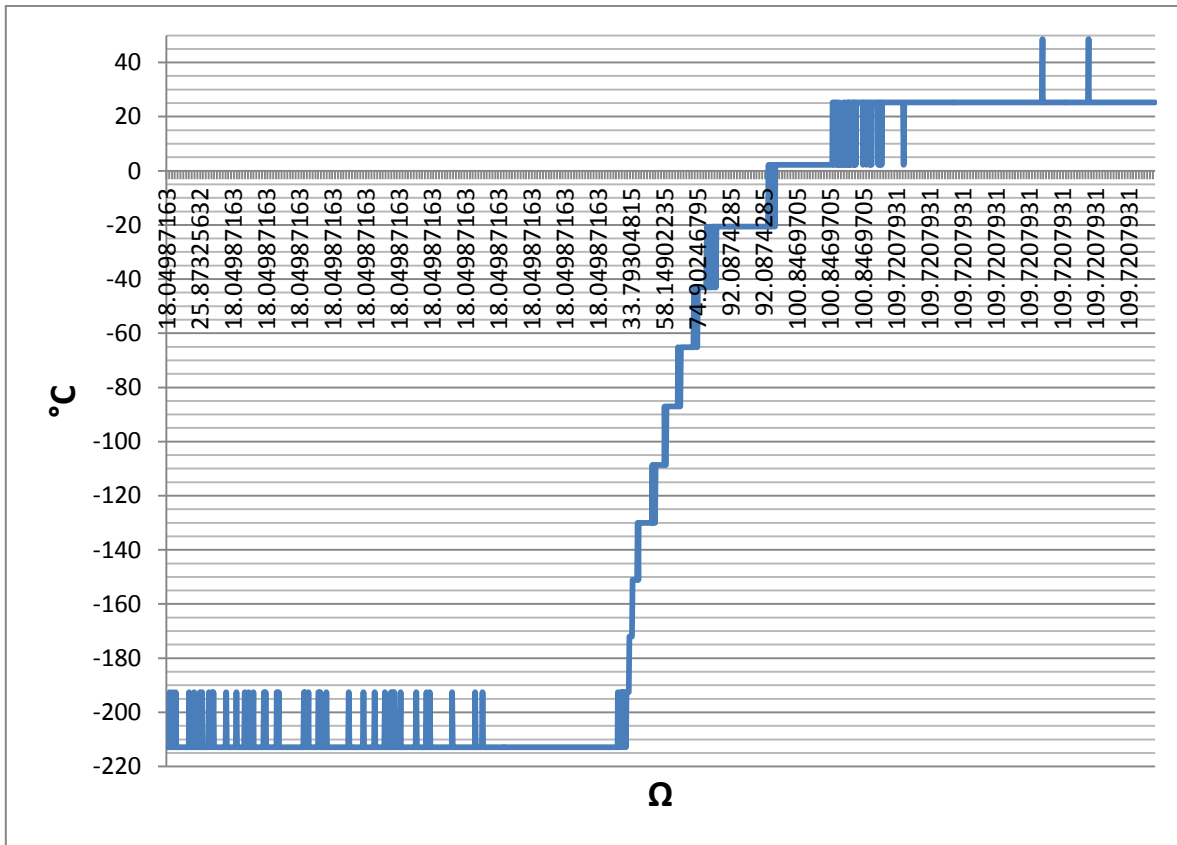


IMAGEN 26 – COMPORTAMIENTO DEL SENSOR CON RESPECTO AL TIEMPO

CONCLUSIONES

La principal limitación de este proyecto es el material de bajo costo, ya que no cuenta con la más alta calidad de semiconductores en su fabricación lo que los hace más susceptibles al ruido y pérdidas de energía.

Los amplificadores utilizados tienen un límite mínimo de voltaje que pueden detectar para amplificar correctamente.

El sensor es de muy lenta respuesta y arrojaba valores muy bajos por lo que se ajustó el modelo del puente para satisfacer el voltaje mínimo ya que no se alcanzaba y el amplificador no operaba como se esperaba. Esto causó que el voltaje diferencial tuviese menor variación, recortando el rango de medición. También al ser amplificadores de una sola fuente carecen de rango de amplificación estable comparado con los de fuente simétrica, de hecho lo ideal es que estos amplificadores sean de instrumentación. Los cuales son los más ideales para las aplicaciones de medición que requieren de mucha precisión.

6. CONCLUSIONES GENERALES

Ambos proyectos resultaron ser satisfactorios dentro de sus limitaciones para haber sido diseñados e implementados con material de bajo costo. Estos diseños solo son prototipos para un modelo a futuro con mejores componentes y mayor eficiencia. La meta fue poderlos operar de manera similar que un instrumento profesional bajo las mismas condiciones para probar su desempeño y tener una mejor idea para un diseño final.

Con esto también se pudo verificar la potencia que posee la *Raspberry Pi* para realizar proyectos que una tarjeta normal no podría. Existen muchas posibilidades más para explotar al máximo la capacidad de esta tarjeta para proyectos de bajo costo y de rápida implementación.

7. REFERENCIAS

Bibliografía

Moreno Velasco, Ignacio. 2013. Sensores Resistivos. Apuntes de Instrumentación Electrónica. Área de Tecnología Electrónica. Universidad Burgos.

Fernández Hermoso, Adoración. 2012. Circuitos con Amplificadores Operacionales. Tema 5

Referencias electrónicas

<http://www.omega.com/prodinfo/rtd.html>

http://www.omega.com/TEMPERATURE/pdf/RTD_GEN_SPECS_REF.pdf

<http://pythonprogramming.net/>

<http://raspimods.blogspot.mx/2014/03/anadiendo-un-adc-la-raspberry-pi.html>

<https://www.raspberrypi.org/>

http://www.electronics-tutorials.ws/opamp/opamp_5.html

APENDICE A

PROGRAMA PARA EL SISTEMA DE ADQUISICIÓN DE DATOS PARA ESTACIÓN METEOROLÓGICA

```
from webiopi.devices.analog import MCP3204
```

```
import wiringpi2
```

```
import datetime
```

```
import os
```

```
import time
```

```
import re
```

```
import math
```

```
def dewpoint(x,y):
```

```
    y1=y*.9
```

```
    y2=y*.1
```

```
    t1=(x/100)
```

```
    t2=math.pow(t1,0.125)
```

```
    dew=round((t2*(112+y1))+(y2-112),2)
```

```
    return dew
```

```
def serialc(port):
```

```
    rv = ""
```

```
    flag = 1
```

```
    try:
```

```

port.write("Q")

while flag == 1:

    try:

        ch = port.read()

        rv += ch

        if ch=='\r' or ch=="":

            flag = 0

    except:

        rv=None

        writefile("Errorlog: ",h+" Unable to read from serial port")

        print "Unable to read from serial port"

    except:

        rv=None

        errorlog("Errorlog: ",h+":Unable to write into serial port")

        print "Unable to write into the serial port"

finally:

    port.close

    print "Original:",rv

    return rv

```

```

def serialp(flujo):

```

```

    try:

        port =serial.Serial(

            flujo,

            baudrate=9600,

```

```
parity=serial.PARITY_NONE,  
stopbits=serial.STOPBITS_ONE,  
bytesize=serial.EIGHTBITS,  
#writeTimeout = 0,  
timeout = 3,  
rtscts=False,  
dsrdtr=False,  
xonxoff=False)  
return port  
except:  
return None
```

```
def UtcNow():  
    U = datetime.datetime.utcnow()  
    return U
```

```
def writefile(name,text):  
    try:  
        log=open(name+f2,"ab+")  
        log.write(text+"\n")  
        log.close()  
    except:  
        print "Unable to write into file"  
    finally:  
        return
```

```
# Setup a MCP3204 on SPI CE0
mcp0 = MCP3204()

port=serialp("/dev/ttyUSB0")

while True: # Loop

    adcsun1 = 0
    adcsun2 = 0
    prom1 = 0
    prom2 = 0

    it = 10

    # Set up date and time
    now = datetime.datetime.now()

    f2=now.strftime("%d-%m-%Y")
    f=now.strftime("%Y-%m-%d")
    h=now.strftime("%H:%M:%S")

    for x in range(0, it):

        adc = mcp0.analogRead(channel=0)
        adc2 = mcp0.analogRead(channel=1)
        volts = adc*.001235351563
        volts2 = adc2*.001235351563
```



```

# Rounds values

voltsr = round(volts,4)
voltsr2 = round(volts2,3)
Vel = volts/.05
Dir = volts2/.0139

# Delay for 100 milliseconds

#wiringpi2.delay(100)

adcsum1 = adcsum1 + Vel
adcsum2 = adcsum2 + Dir

prom1 = round(adcsum1/it,1)
prom2 = round(adcsum2/it,0)

# Rounds Values

promr1 = round(prom1,2)
promr2 = round(prom2,2)

print "-----"
print "Current Date:",f2
print "Current Time:",h
print "UTC Date time:",UtcNow().strftime("%d-%m-%Y, %H:%M:%S")
print
print ("Speed: {} m/s ({}V)".format(promr1,voltsr))
print
print ("Direction: {} deg ({}V)".format(promr2,voltsr2))
print "-----"

```

```

response=serialc(port)

if port is not None:
    try:
        l=re.sub(r"[KHC%m/bF]", "", response)
        w=l.split()
        lw= len (w)
        print "Port:",port
        print "Clean:",l
    except:
        lw=0
        writefile("Errorlog: ",h+" Error while reading data")
        print "Error while reading data"

    if (lw==10):
        x=float(w[7])
        y=float(w[5])
        print "DP=", dewpoint(x,y)

        writefile("Datalog: ", UtcNow().strftime("%d-%m-%Y,%H:%M:%S")+","+now.strftime("%d-%m-%Y,%H:%M:%S")+","+str(promr1)+","+str(voltsr)+","+str(promr2)+","+str(voltsr2))

    else:
        writefile("Errorlog: ",h+" Error while reading data; serial port may be disconnected or data may be corrupted")
        print "Error while reading data; serial port may be disconnected or Data may be corrupted"

```

else:

```
writefile("Errorlog: ",h+" Error: No port detected")
```

```
print "Error: No port detected"
```

Wait for x seconds

```
time.sleep(10)
```

APENDICE B

PROGRAMA PARA EL SISTEMA DE ADQUISICIÓN DE DATOS PARA SENSOR CRIOGÉNICO

```
import spidev
import time
import math
import datetime

class MCP3204:

    def __init__(self, spi_channel=0):

        self.spi_channel = spi_channel

        self.conn = spidev.SpiDev(0, spi_channel)

        self.conn.max_speed_hz = 1000000 # 1MHz

    def __del__( self ):

        self.close

    def close(self):

        if self.conn != None:

            self.conn.close

            self.conn = None

    def bitstring(self, n):

        s = bin(n)[2:]

        return '0'*(8-len(s)) + s
```

```

def read(self, adc_channel=0):
    # build command
    cmd = 128 # start bit
    cmd += 64 # single end / diff
    if adc_channel % 2 == 1:
        cmd += 8
    if (adc_channel/2) % 2 == 1:
        cmd += 16
    if (adc_channel/4) % 2 == 1:
        cmd += 32

    # send & receive data
    reply_bytes = self.conn.xfer2([cmd, 0, 0, 0])

    #
    reply_bitstring = ''.join(self.bitstring(n) for n in reply_bytes)
    # print reply_bitstring

    # see also... http://akizukidenshi.com/download/MCP3204.pdf (page.20)
    reply = reply_bitstring[5:19]
    return int(reply, 2)

if __name__ == '__main__':
    spi = MCP3204(0)

    count = 0

```

```
a0 = 0
```

```
a1 = 0
```

```
a2 = 0
```

```
a3 = 0
```

```
while True:
```

```
    count += 1
```

```
    a3 += spi.read(3)
```

```
    now = datetime.datetime.now()
```

```
    f=now.strftime("%d-%m-%Y")
```

```
    h=now.strftime("%H:%M:%S")
```

```
    if count == 10:
```

```
        Vo((((a3/10)*3.3)/4096))+0.002
```

```
        Ro=(182.60113-(1000*round(Vo,3)))/(round(Vo,3)+.027298869)
```

```
        Rtd=Ro-259.5
```

```
        T=((Rtd/100)-1)/.00385
```

```
        K=T+273
```

```
        print f, h, "Vo =%0.3f V, Rtd =%0.4f Ohm, T =%0.4f C, K =%0.4f K" % (Vo, Rtd, T, K)
```

```
        count = 0
```

```
        a0 = 0
```

```
        a1 = 0
```

```
a2 = 0
a3 = 0
file = open(f,"a")
file.write("{} V, {} Ohm, {} C, {} K\n".format(f, h, Vo, Rtd, T, K))
# Wait for x seconds
time.sleep(1)
```