

ADQUISICIÓN DE DATOS A TRAVÉS DE COMUNICACIÓN SERIAL UTILIZANDO EL MÓDULO DE RADIO TELEMETRÍA 3DR Y PYTHON

M. en C. Esther Viridiana Vázquez Carmona
evazquezc1801@alumno.ipn.mx
M. en C. Rodrigo Vázquez López
rvazquezl1800@alumno.ipn.mx
M. en C. Luis Alberto Flores Montaña
luisfloresmontano@hotmail.com
Dr. Juan Carlos Herrera Lozada
jlozada@ipn.mx

Instituto Politécnico Nacional
Centro de Innovación y Desarrollo Tecnológico en
Cómputo #

Resumen

Actualmente los sistemas de monitoreo buscan atender necesidades para tomar decisiones en tiempo real y coadyuvar en la optimización de procesos. Este documento retoma el desarrollo de un sistema de monitoreo que pretende atacar problemas climáticos como heladas, lluvias y golpes de calor que pueden afectar un entorno, integrando ahora la etapa de adquisición de datos a través de un módulo de radio telemetría. Los datos se obtienen de los sensores DHT-22, BMP-180 y NEO-6M, posteriormente son enviados al puerto serie de una microcomputadora que transmite los datos por medio de una de las antenas de telemetría, estos datos son recibidos por una antena receptora que se encuentra conectada a una computadora. El propósito de esta etapa es enviar los datos de forma inalámbrica de manera que el sistema de monitoreo sea móvil y que la etapa de procesamiento se realice en la central de monitoreo. Como resultado los datos de este sistema de monitoreo son enviados vía inalámbrica hacia una computadora a través de los puertos serie de cada dispositivo y la adquisición de datos se realiza vía remota.

1. Introducción

El desarrollo de los sistemas de monitoreo ha permitido a las industrias y otros sectores la optimización de recursos, observando variables que se puedan producir en el tiempo, algunos sistemas de monitoreo llamados SCADA se basan en computadoras que permiten supervisar y controlar a distancia dichas variables (Suhissa, s.f.). Cabe destacar que este trabajo integra la parte de adquisición de datos a través de un módulo de telemetría, el trabajo previo que se está retomando es un prototipo de un sistema de monitoreo de variables climáticas (Vázquez et al., 2019) en los que intervienen sensores de humedad, presión, temperatura y un sensor GPS para conocer la ubicación a través de la latitud y longitud durante la recolección de datos.

1.1 Adquisición de datos

La adquisición de datos consiste en tomar muestras de variables físicas como temperatura, corriente, voltaje que puedan ser manipuladas y digitalizarlas para su procesamiento. Esta adquisición se realiza a través de un sensor o transductor que tomará los datos vía puerto serial.

1.2 Comunicación serial

La comunicación serial es el intercambio de información transmitiendo un bit a la vez utilizando la interfaz USB (Bus Serie Universal), SPI (Interfaz Periférica Serial), I²C (Circuito Inter-integrado), Ethernet o la interfaz estándar TX/RX. Existen dos tipos de interfaces: síncronas y asíncronas. Una interfaz síncrona se acompaña de una señal de reloj, dentro de estas interfaces se encuentran SPI e I²C. En una interfaz asíncrona los datos se transfieren sin una señal de reloj para ello se utilizan una serie

de parámetros para garantizar la transferencia de información como: los bits de datos (Oscila entre 5 -9 bits en este parámetro también se especifica el formato de datos puede ser MSB O LSB), bits de sincronización (Son 2 o 3 bits que se encargan de especificar el inicio y el final de un paquete de datos), bits de paridad (es una forma de comprobar errores puede ser impar o par sumando los bits de byte de los datos) y velocidad de baudios (velocidad en la cual se enviaran los datos, por lo general se maneja una velocidad de 9600 bps para no comprometer los datos). Estos datos se manipulan fácilmente con la ayuda de la biblioteca pySerial en Python (Didácticos Ecarletti, 2019).

1.3 Biblioteca pySerial y time

Este módulo se encarga de abrir acceso al puerto serie a partir de los parámetros antes mencionados agregando el número de puerto, la velocidad de los datos, el número de bytes, entre otros. El puerto se abre inmediatamente cuando se proporciona un puerto y se crean los objetos, además requiere una llamada sucesiva para abrir. El puerto va a depender del sistema operativo por ejemplo en Windows suelen llamarse COM# (en administrador de dispositivos se encuentra el número que corresponde al puerto COM que se intenta leer) y en GNU/Linux se agrega la ruta de dicho puerto (/dev/ttyUSB# , para saber el número de puerto en el que se encuentra conectado el dispositivo que brindará los datos en este caso los sensores desde la placa Arduino con el comando `dmesg | grep tty`). Por otro lado, la biblioteca time se encarga de realizar retardos durante la transmisión de datos (Liechti, 2015).

1.4 Telemetría

Es una tecnología utilizada para la medición y monitoreo de forma remota de variables físicas o químicas. El sistema de telemetría se lleva a cabo de manera inalámbrica y por otros medios como computadoras o teléfonos. La telemetría funciona a través de un transductor como el dispositivo de entrada, un medio transmisor que emite ondas de radio. El transductor convierte la magnitud física en una señal eléctrica que es transmitida a una antena de recepción (en este caso) (Cengage, 2016).

2. Desarrollo

Se desarrolló un prototipo previo que incluye sensores de variables climáticas, integrado por sensores de humedad, presión barométrica que funcionan en conjunto con un sensor GPS para conocer la ubicación. Sin embargo, estos datos llegaban solo a una computadora, la diferencia es que la placa Arduino se encuentra conectada a una microcomputadora Raspberry Pi 3 y los datos se envían a través de una antena de radio telemetría. El diseño del hardware se muestra a continuación.

2.1 Diseño del hardware

El hardware está integrado por los sensores DHT-22 para humedad, BMP-180 para presión barométrica y NEO-6M para GPS a través de la latitud y longitud. Dichos sensores están interconectados hacia una placa Arduino nano. La placa a través de su interfaz USB es interconectada a una de las entradas USB de la microcomputadora Raspberry 3 utilizando el mismo cable de alimentación de dicha placa, posteriormente la antena emisora de radio telemetría es conectada a la microcomputadora a través de un módulo FTDI hacia una de las entradas USB de la microcomputadora, este módulo se utiliza entre estos dispositivos ya que el voltaje que manejan las antenas de telemetría es diferente al que maneja la Raspberry 3. La Raspberry Pi 3 trabaja con 3.3 V mientras que las antenas con 5.0 V. El trabajo de este módulo es convertir el voltaje proveniente de la microcomputadora a un voltaje adecuado para la alimentación de la antena emisora, cabe destacar que este módulo también hace la conversión de USB a TTL UART que es la interfaz que utiliza dichas antenas (Future Technology Devices International Limited, 2009). En la figura 1 se muestra el diseño de hardware para el envío de datos.

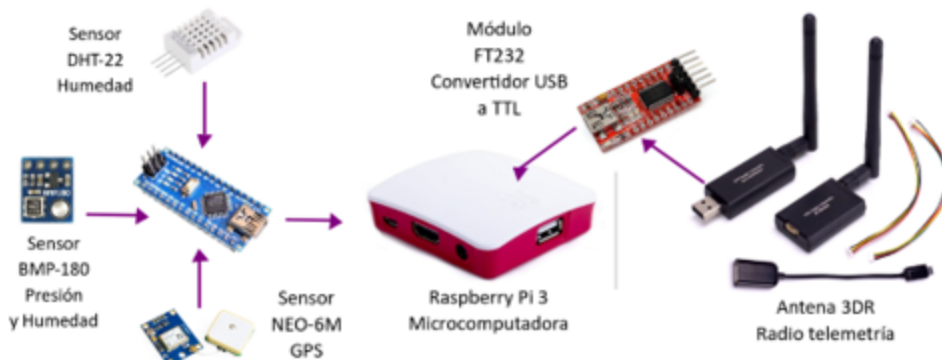


Figura 1. Integración del prototipo de variables climáticas a una microcomputadora para envío de datos de forma inalámbrica.

Posteriormente los datos de los sensores son enviados a través de la antena emisora de radio telemetría hacia una antena receptora que tiene un alcance aproximadamente de 1609.34 metros (Roboticspk, 2014). La figura 2 muestra el hardware que realiza la adquisición de datos.



Figura 1. Recepción de datos o adquisición de datos.

2.2 Diseño de Software

Para leer los datos vía serial de la placa Arduino se creó un script en Python que contiene los parámetros para iniciar el funcionamiento de la antena emisora. A continuación, en la figura 3 se describe el código.

```

1 import serial      # Biblioteca que da acceso al puerto serial
2 import time        # Biblioteca para sincronizar la hora del sistema
3 import numpy as np # Biblioteca para operar con matrices y vectores

1
2 datos = serial.Serial( port="/dev/ttyUSB0",      # Puerto que se va a leer, en este caso donde se encuentra
3                       baudrate=9600,          # conectada la placa Arduino en la microcomputadora GNU/Linux
4                       parity=serial.PARITY_NONE, # Sin paridad
5                       stopbits=serial.STOPBITS_ONE, # Bits de parada
6                       bytesize=serial.EIGHTBITS, # Tamaño de byte 8 bits
7                       timeout=1)              # Tiempo de espera por cada solicitud
8
9
10
11 enviar=serial.Serial( port="/dev/ttyUSB1",      # Puerto donde se encuentra la antena emisora
12                       baudrate=57600,         # Velocidad con la que trabajan las antenas

1
2 miCadena=[]      # Declarar variable tipo arreglo o array
3
4 while True:
5     time.sleep(2) # Tiempo en que se ejecutará el siguiente bloque de código
6     rawString=datos.readline().decode("utf-8") # Lee una línea completa de datos desde el puerto de la placa Arduino
7     # con el formato utf-8.
8     miCadena.append(rawString.split(",")) # Guarda los datos en un arreglo, eliminando caracteres que
9     # se encuentran en la cadena a través del método split().
10    cadena_enviar=(bytes(rawString,'utf-8')) # Cambiar el formato de bytes para realizar envío
11    print(cadena_enviar) # Imprime datos que se van a enviar
12    sendString=enviar.write(cadena_enviar) # Envío de datos a la antena receptora con el método write()
13

```

Figura 3. Script en Python para leer datos de Arduino y realizar el envío a través de la antena emisora de radio telemetría.

Para recibir los datos de la antena receptora, se utilizó un programa cliente (Tatham, 2019) cuya función es el control de dispositivos serie a través del acceso remoto hacia un canal de la computadora. La figura 4 muestra la configuración de cada parámetro.

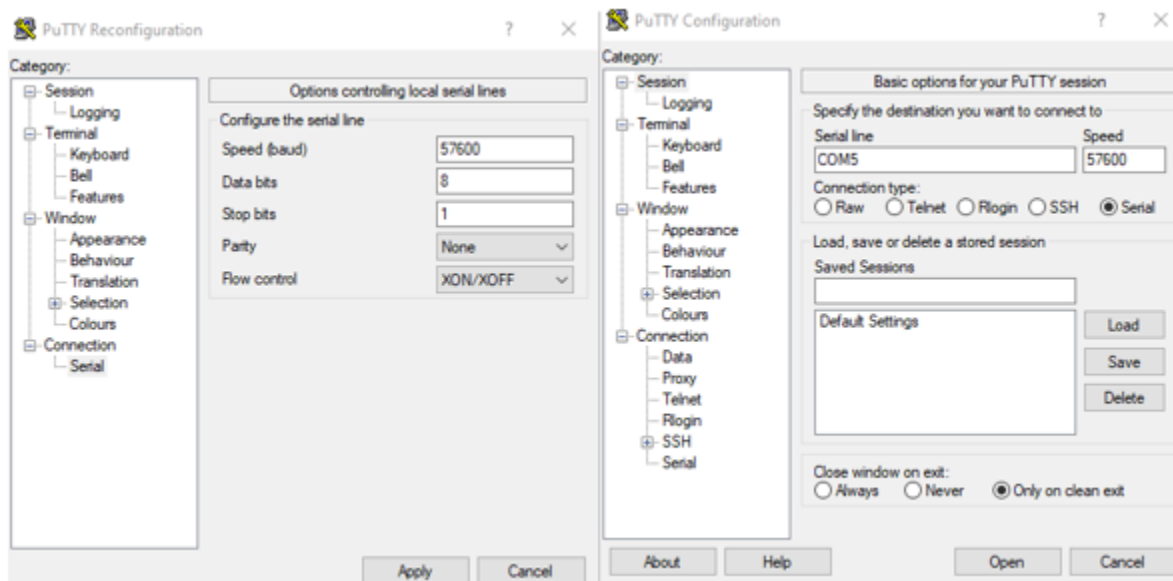


Figura 4. Parámetros para abrir una sesión de tipo serial.

En la figura 5 se puede observar la implementación del hardware. Como se puede observar los datos de los sensores llegan a la microcomputadora Raspberry Pi 3 a través de la placa Arduino, enseguida son transmitidos por la antena emisora.

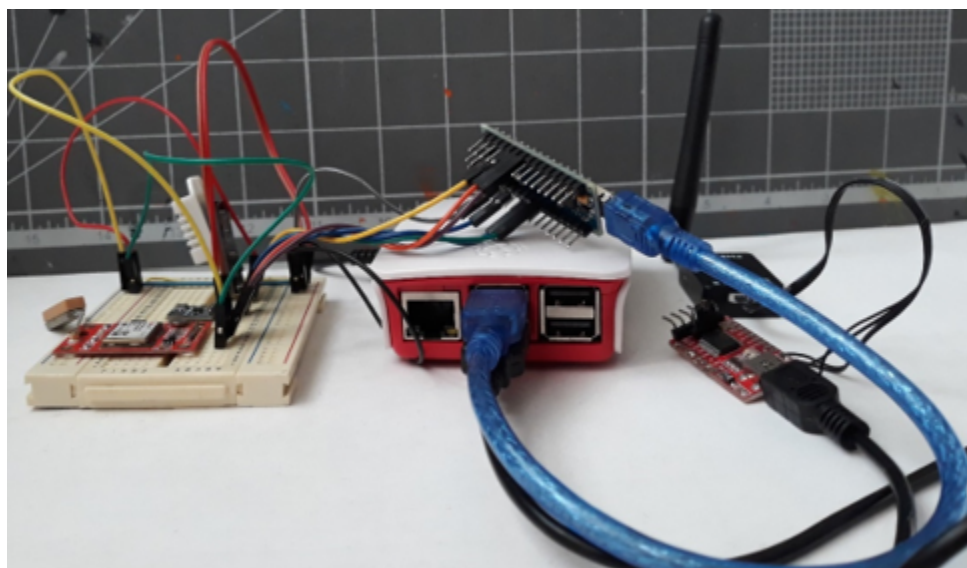


Figura 5. Implementación de prototipo con la antena de radio telemetría emisora.

3. Resultados

Los datos de los sensores son enviados a la microcomputadora, se muestran en la figura 6. Los datos recibidos por la antena receptora se visualizan en la figura 7.

```
b' Humedad, Temperatura, Presion, Altitud, Latitud, Longitud, Fecha, Hora, \r\n'
b' 24.90, 34.72, 779.33, 2159.85, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
b' 25.00, 34.72, 779.33, 2159.87, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
b' 25.10, 34.73, 779.34, 2159.76, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
b' 25.30, 34.75, 779.39, 2159.21, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
b' 25.60, 34.75, 779.33, 2159.82, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
b''
b' nan, 34.76, 779.36, 2159.51, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
b' 24.20, 34.76, 779.36, 2159.58, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
b' 25.20, 34.72, 779.42, 2158.92, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
b' 25.00, 34.68, 779.33, 2159.83, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
b''
b' 24.90, 34.70, 779.36, 2159.50, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
b' 24.90, 34.75, 779.41, 2159.04, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
b' 24.90, 34.76, 779.34, 2159.77, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
b''
b' 25.00, 34.73, 779.35, 2159.64, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
b' 25.00, 34.78, 779.33, 2159.89, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
b' 25.20, 34.79, 779.37, 2159.47, 0.00, 0.00, 0/0/2000, 0:0:0, \r\n'
```

Figura 6. Datos que se obtienen de los sensores vistos desde la microcomputadora. Cabe destacar están en formato utf-8.

COM5 - PuTTY

```
Humedad, Temperatura, Presion, Altitud, Latitud, Longitud, Fecha, Hora,  
24.90,34.72,779.33,2159.85,0.00,0.00,0/0/2000,0:0:0,  
25.00,34.72,779.33,2159.87,0.00,0.00,0/0/2000,0:0:0,  
25.10,34.73,779.34,2159.76,0.00,0.00,0/0/2000,0:0:0,  
25.30,34.75,779.39,2159.21,0.00,0.00,0/0/2000,0:0:0,  
25.60,34.75,779.33,2159.82,0.00,0.00,0/0/2000,0:0:0,  
nan,34.76,779.36,2159.51,0.00,0.00,0/0/2000,0:0:0,  
24.20,34.76,779.36,2159.58,0.00,0.00,0/0/2000,0:0:0,  
25.20,34.72,779.42,2158.92,0.00,0.00,0/0/2000,0:0:0,  
25.00,34.68,779.33,2159.83,0.00,0.00,0/0/2000,0:0:0,  
24.90,34.70,779.36,2159.50,0.00,0.00,0/0/2000,0:0:0,  
24.90,34.75,779.41,2159.04,0.00,0.00,0/0/2000,0:0:0,  
24.90,34.76,779.34,2159.77,0.00,0.00,0/0/2000,0:0:0,  
25.00,34.73,779.35,2159.64,0.00,0.00,0/0/2000,0:0:0,  
25.00,34.78,779.33,2159.89,0.00,0.00,0/0/2000,0:0:0,  
25.20,34.79,779.37,2159.47,0.00,0.00,0/0/2000,0:0:0,  
25.30,34.78,779.36,2159.58,0.00,0.00,0/0/2000,0:0:0,  
25.30,34.79,779.38,2159.32,0.00,0.00,0/0/2000,0:0:0,  
25.40,34.81,779.32,2159.96,0.00,0.00,0/0/2000,0:0:0,  
nan,34.82,779.37,2159.46,0.00,0.00,0/0/2000,0:0:0,  
24.20,34.81,779.37,2159.50,0.00,0.00,0/0/2000,0:0:0,  
24.90,34.81,779.32,2159.97,0.00,0.00,0/0/2000,0:0:0,  
24.90,34.82,779.35,2159.61,0.00,0.00,0/0/2000,0:0:0,  
24.80,34.84,779.39,2159.26,0.00,0.00,0/0/2000,0:0:0,
```

Figura 7. Datos de la antena receptora de radio telemetría. Nótese que los caracteres utf-8 no aparecen cuando ya son enviados, estos caracteres son interpretados por el programa.

Conclusiones

Se desarrolló previamente un prototipo sencillo que monitorea variables como presión, temperatura, humedad, así como la localización de dicho prototipo a través de longitud y latitud, en el que se incluye la hora y fecha de cada toma. En este prototipo se integra una microcomputadora y un módulo de radio telemetría para que los datos sean enviados de forma inalámbrica. Los sensores se interconectan a una placa Arduino que a su vez por medio de la UART que se encuentra en el puerto USB es interconectado hacia un puerto USB de la microcomputadora Raspberry Pi 3. La microcomputadora envía los datos hacia una antena emisora de radio telemetría. Posteriormente una PC que tiene conectada la antena receptora, recibe los datos y se visualizan con la ayuda de un programa con acceso al puerto serie. La siguiente etapa del prototipo consistirá en almacenar los datos de cada variable sin la necesidad de utilizar un programa para el control de dispositivos del puerto serial, sino a través de un Script de Python.

Referencias

1. Cengage. (2019, 20 junio). *¿Qué es la telemetría?* Cengage. Recuperado 25 marzo, 2020, de <https://latinoamerica.cengage.com/que-es-la-telemetria/>
2. Ecarletti, R. Didácticos. (2019, 20 junio). *¿Qué es la comunicación serie?* Robots Didácticos. Recuperado 25 marzo, 2020, de <http://robots-argentina.com.ar/didactica/que-es-la-comunicacion-serie/>
3. Future Technology Devices International Limited. (2009). *FTDI Chip Home Page [DataSheets FT232R]. libro, revista o nombre de la página web* Recuperado 25 marzo, 2020, de <https://www.ftdichip.com>

4. Liechti, C. (2015). *Welcome to pySerial's documentation* pySerial 3.0 documentation. Recuperado 25 marzo, 2020, de <https://pythonhosted.org/pyserial/>
5. Roboticspk. (2014). *RoboticsPK Pakistan's Own Robotics and Electronics Store - ROBOTICS PK [Conjunto de datos]*. Recuperado 25 marzo, 2020, de <http://www.roboticspk.com/index.php?tmpl=component>
6. Suhissa. (s.f.). *Sistemas de monitoreo* Suhissa. Recuperado 25 marzo, 2020, de <https://suhissa.com.mx/sistemas-de-monitoreo/>
7. Tatham, S. (2019). *puTTY Documentation Page* [App de móvil]. Recuperado 25 marzo, 2020, de <https://www.chiark.greenend.org.uk/~%7Esgtatham/putty/docs.html>
8. Vázquez, E. V., Vázquez, R., & Herrera, J. C. (2019, 1 septiembre). *Arduino: monitoreo de variables climáticas haciendo uso de los sensores bmp-180, dht-22 y neo-6m*. Boletín UPIITA, -(74), 1-7. Recuperado de <http://www.boletin.upiita.ipn.mx/>