Monitoreo y Control Industrial Usando Python

PyCon Argentina 2010 Córdoba

Autor: Joaquín Sorianello <soriasoft@gmail.com>

Fecha: 16/10/2010

En python

```
#!-*- coding: utf8 -*-
"""Cliente serie para la balanza nc3m"""

import struct
import serial
import decimal

def decimal_from_nc3m(nc3m_num):
    """Toma un numero en el formato NC3M y lo convierte a decimal"""
    nc3m_num = nc3m_num.replace(',', '.')
    return decimal.Decimal(nc3m_num)

def main():
    #definimos el string de formato
    fcn = 'c8sc7s2c'
    #creamos una conexión serie
    ser = serial.Serial('/home/joac/COM9')
    totalizador = 0
    #Adquirimos los datos
    while True:
```

```
        a = ser.readline() #Leemos una linea del buffer
        stx, neto, status, tara, cr, lf = struct.unpack(fcn, a)
```

```
        if status == ' ':  #Chequeamos que la balanza esté en equilibrio
            neto = decimal_from_nc3m(neto)
            totalizador += neto
            print "Peso Neto: %s Peso Acumulado: %s" % ( neto, totalizador)

if __name__ == "__main__":
    print "Cliente serie para balanza NC3M"
    main()
```

ModbusTk

ModbusTk, es un toolkit para comunicarse con dispositivos de campo, utilizando el protocolo Modbus, ya sea RTU o TCP/IP y para crear dispositivos virtuales (Muy útil para realizar mockups)

Como funciona Modbus (en forma muy general)

Modbus tiene una arquitectura Maestro-Esclavo, donde un único dispositivo Maestro recoge datos y establece parámetros en los dispositivos Esclavos. Establece en los dispositivos cuatro tipos de registros:

- **Discretas**
  - Solo lectura
  - lecto-escritura
- **Analógicas**
  - solo lectura
  - lecto-escritura.

Ademas establece códigos de funciones, para realizar operaciones en dichos registros

Ventajas

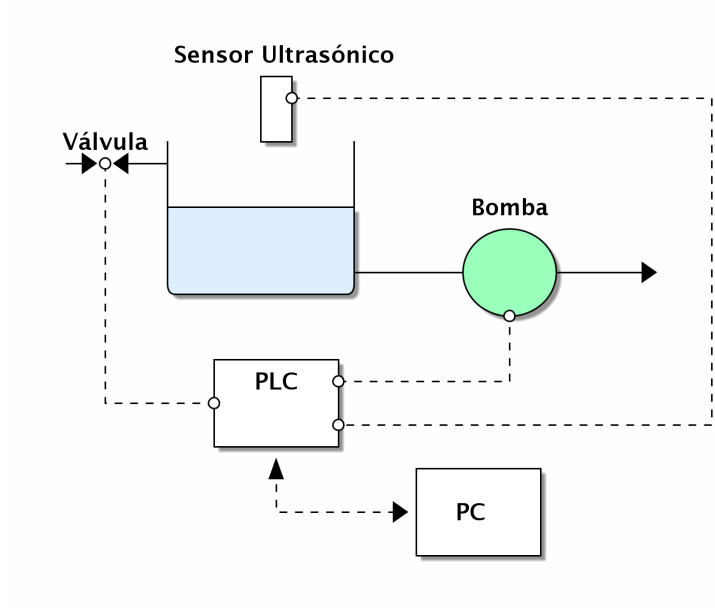
- El protocolo Modbus es abierto y esta completamente documentado.
- En Modbus/RTU podemos tener muchos dispositivos en el mismo bus.
- Existen conversores de ModbusRTU en ModbusTCP/IP
- Tiene control de errores.
- No depende de la plataforma
- ¿Ya dije que es un protocolo abierto?

Desventajas

- Muchos PLC (Siemens, por ejemplo) y dispositivos de gama baja no lo implementan.
- ModbusRTU no soporta Mulimaster.

Ejemplo

Tanque con sensor ultrasónico, una válvula y una bomba, gobernado por un PLC



OpenOPC

Es un toolkit OPC-DA para python.

Que es OPC?

Es el acrónimo para Object Linking and Embedding (OLE) for Process Control. Es un estándar que permite la comunicación en tiempo real entre aplicaciones de distintos fabricantes. Los datos se obtienen a través de *Servidores OPC* Hay varias versiones, pero la mas utilizada es OPC-DA (fuertemente atada a Window$, ya que utiliza DCOM)

Ventajas

- No tenemos que preocuparnos en la comunicación explicita con los dispositivos.
- Es sencillo de utilizar.
- Podemos acceder a muchos dispositivos con diversos protocolos, con una interfaz común.
- Es la única forma (estable) que encontré para comunicarme con dispositivos Siemens de gama media/baja.

- OpenOPC puede ser utilizado para acceder de forma remota a servidores OPC utilizando PyRO

Desventajas

- Los Servidores suelen ser pagos (y bastante caros)
- Necesitamos un equipo con windows

Ejemplo

TODO

Otros módulos de comunicaciones

Porque Python

Su gran cantidad de modulos:

- **Toolkits Graficos**
  - PyQt
  - PyGTK
  - WxPython
- **Herramientas para Graficación:**
  - MatplotLib
- **ORMs**
  - Sql Alchemy
  - Elixir

- **Frameworks Web**
  - Django
  - Bottle
- Twisted

Que permiten crear soluciones sofisticadas e innovadoras en materia de supervisión y control industrial

¿Preguntas?