FACULTAD DE INGENIERIA
Universidad de Buenos Aires

Protocolos de comunicación

Especialización en Sistemas Embebidos

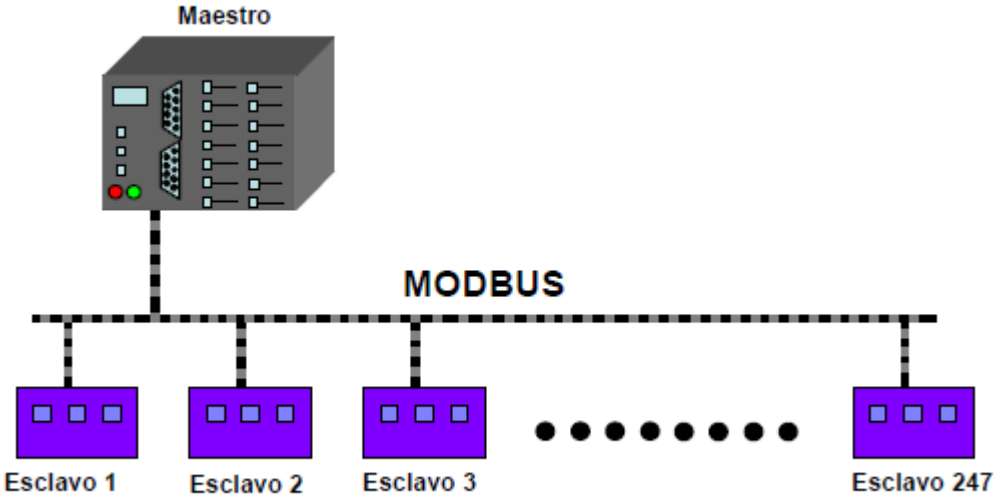
Sensor de múltiples variables con protocolo Modbus

Autor: Ing. Lucas Dórdolo

Docentes:
Esp. Ing. Eric Pernia
Esp. Ing. Pablo Gómez

Protocolo Modbus

- Simple
- Muy difundido
- Apto ambientes industriales
- Económico
- Escalable



Protocolo Modbus

The following figure gives a general representation of MODBUS serial communication stack compared to the 7 layers of the OSI model.

Layer	ISO/OSI Model	
7	Application	MODBUS Application Protocol
6	Presentation	Empty
5	Session	Empty
4	Transport	Empty
3	Network	Empty
2	Data Link	MODBUS Serial Line Protocol
1	Physical	EIA/TIA-485 (or EIA/TIA-232)

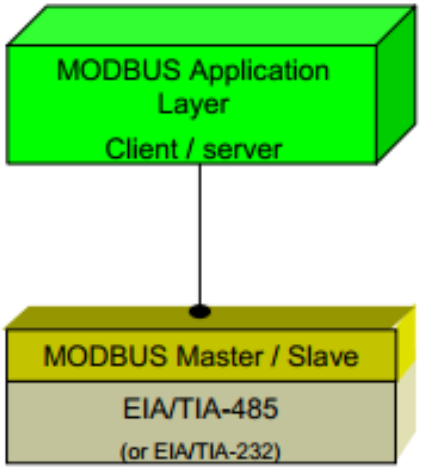
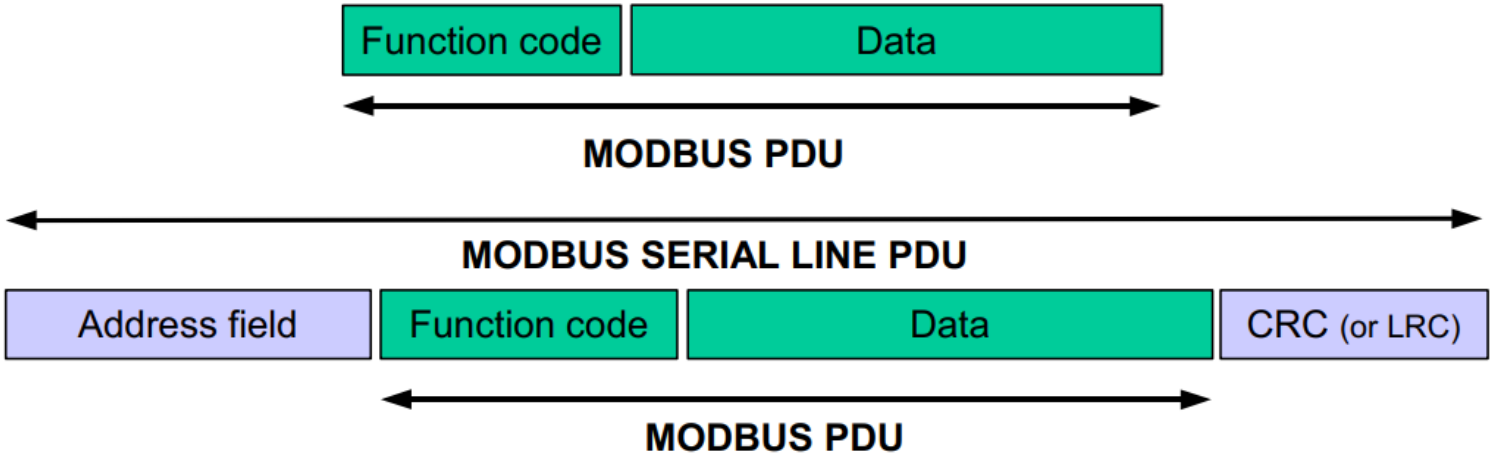


Figure 2: MODBUS Protocols and ISO/OSI Model

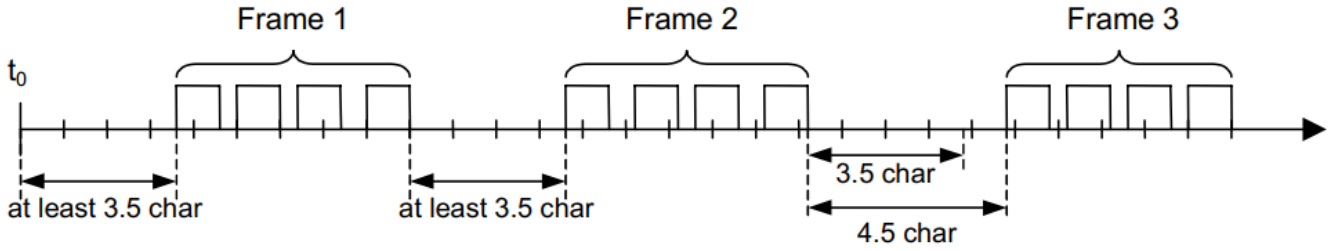
Protocol Data Unit (PDU)



Modbus: funciones más comunes

Función	Código	Descripción
Read Coils	1	Leer estado de salidas discretas (en general el estado de los reles).
Read Discrete Inputs	2	Leer estado de entradas discretas.
Read Holding Register	3	Leer valores de registros “Holding”.
Write Single Coil	5	Permite modificar el valor de una sola salida discreta.
Write Register	6	Escribe un valor en un registro.
Write Multiple Coils	15	Permite modificar el valor de múltiples salidas discretas al mismo tiempo.
Write Multiple Registers	16	Escribe múltiples registros al mismo tiempo.

Modbus RTU



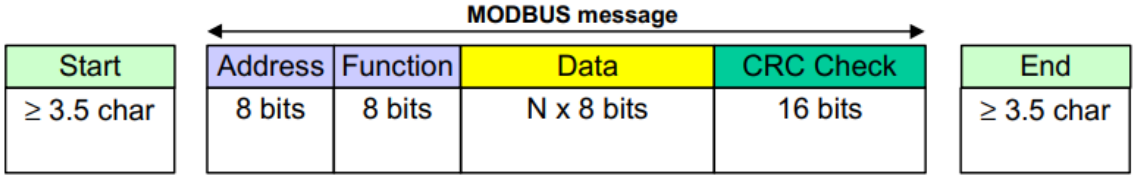
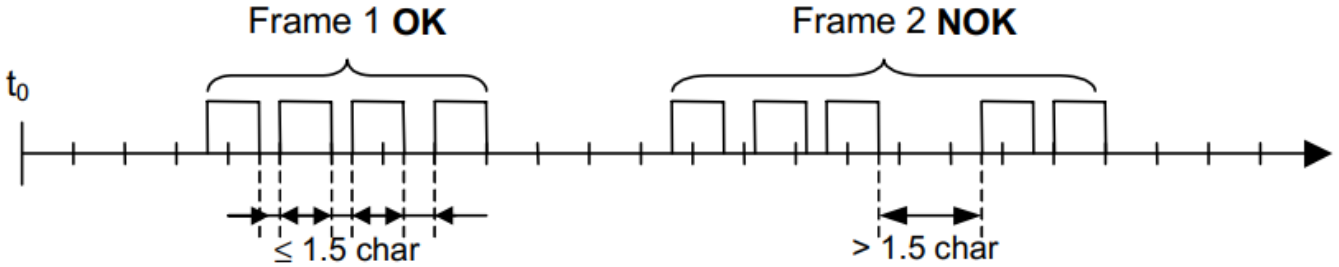


Figure 13: RTU Message Frame



Modbus ASCII

Timeout de 1 segundo entre caracteres

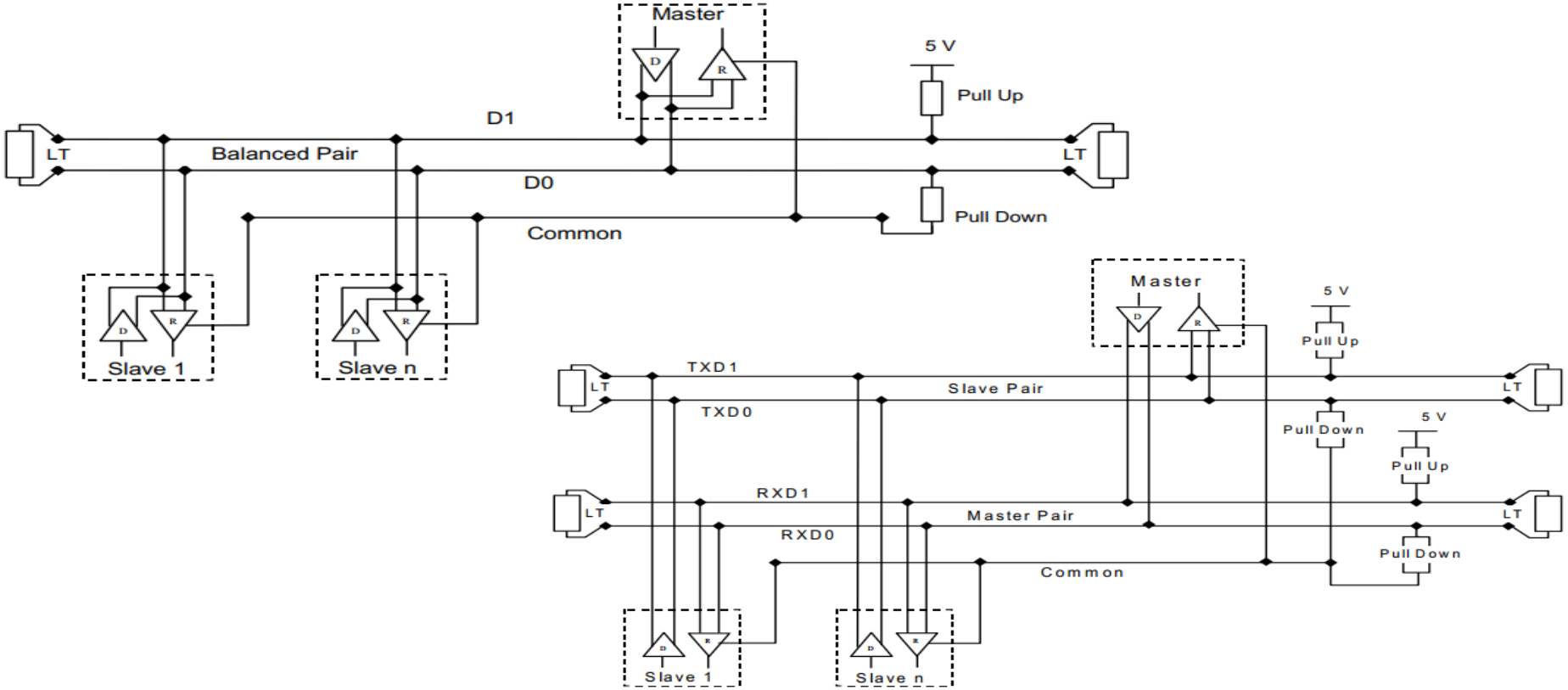
Start	Address	Function	Data	LRC	End
1 char :	2 chars	2 chars	0 up to 2x252 char(s)	2 chars	2 chars CR,LF

Comparación ASCII / RTU

Properties of Modbus/ASCII and Modbus/RTU

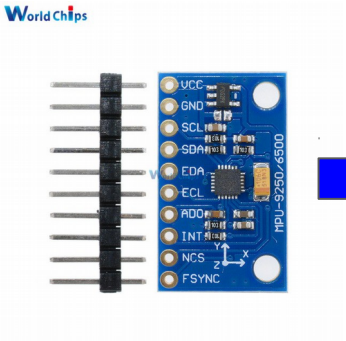
	Modbus/ASCII		Modbus/RTU	
Characters	ASCII **0**...**9** and **A**..**F**		Binary **0**...**255**	
Error check	**LRC** Longitudinal Redundancy Check		**CRC** Cyclic Redundancy Check	
Frame start	character ':'		3.5 chars silence	
Frame end	characters **CR**/**LF**		3.5 chars silence	
Gaps in message	1 sec		1.5 times char length	
Start bit	**1**		**1**	
Data bits	**7**		**8**	
Parity	even/odd	none	even/odd	none
Stop bits	**1**	**2**	**1**	**2**

RS-485 sobre 2 y 4 cables



Esquema de conexiones

MPU9250



I2C

EDU-CIAA NXP

MODBUS RTU
RS-485

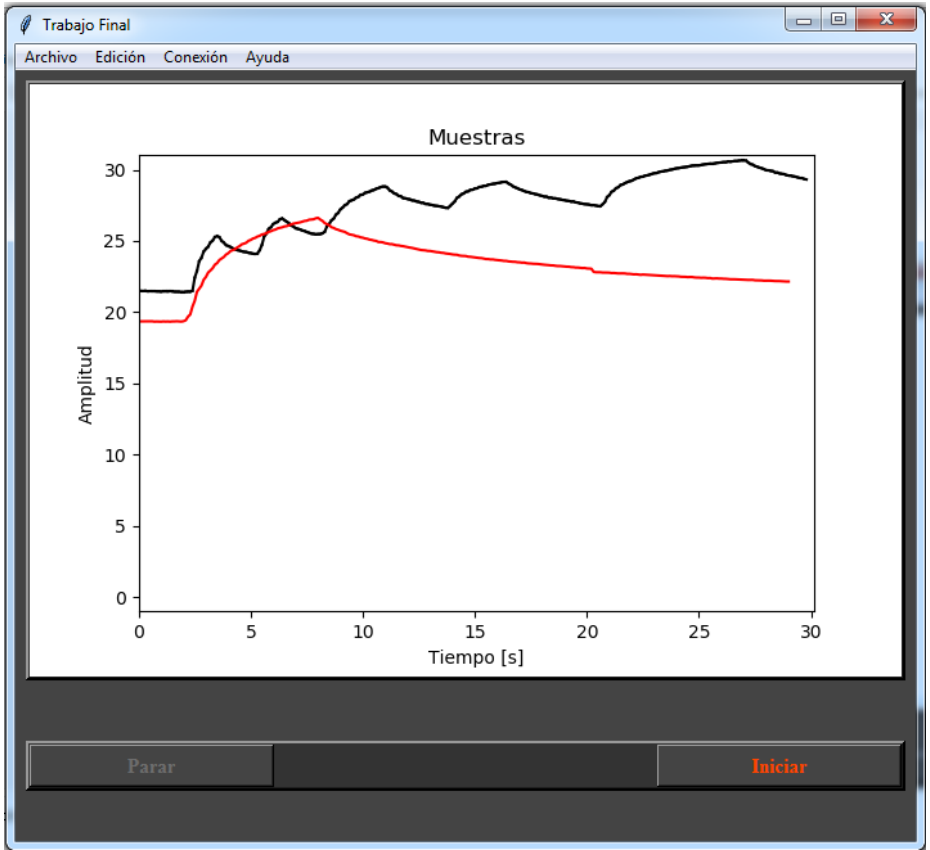
PC Master

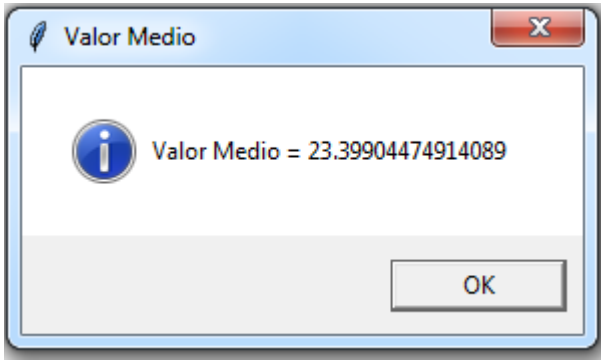
Lectura de Registros

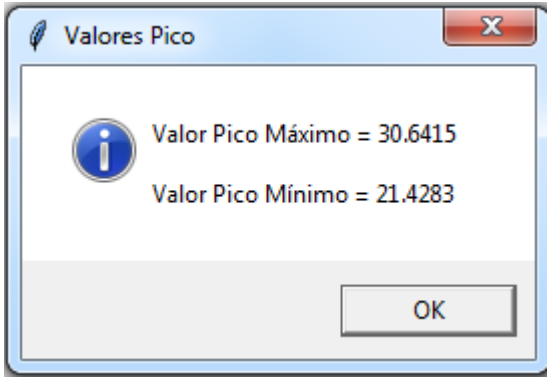
```
#Connect to the slave
PORT = test_comm.selec_comm('USB-SERIAL CH340')
master = modbus_rtu.RtuMaster(
    serial.Serial(port=PORT, baudrate=115200, bytesize=8, parity='N', stopbits=2, xonxoff=0)
)
master.set_timeout(5.0)
master.set_verbose(True)
logger.info("connected")

lista = list(master.execute(10, cst.READ_HOLDING_REGISTERS, 0, 20 ))

tmp=numpy.array(lista, numpy.int16)
tmp.dtype = numpy.float32

logger.info(tmp)
```

```
2018-08-24 02:13:57,020 INFO    modbus_rtu.__init__      MainThread      RtuMaster COM7 is opened
2018-08-24 02:13:57,038 INFO    modbus_master.main       MainThread      connected
2018-08-24 02:13:57,041 DEBUG   modbus.execute  MainThread      -> 10-3-0-0-0-20-68-190
2018-08-24 02:13:57,059 DEBUG   modbus.execute  MainThread      <- 10-3-40-159-134-65-153-45-44-187-31-200
-128-57-194-53-76-186-113-177-240-65-6-221-57-64-2-52-125-192-161-128-10-65-63-199-174-65-175-134-222-194-
103-132-17
2018-08-24 02:13:57,064 INFO    modbus_master.main       MainThread      [  1.92028923e+01  -2.42883991e-03
  3.71519476e-04  -9.20136226e-04
   8.41844177e+00   2.04475236e+00  -5.03765726e+00   1.19687595e+01
   2.19724998e+01  -5.78817062e+01]
```

Capturas Varias







Muchas Gracias!!