**MODBUS**:

Il Modbus è essenzialmente un protocolo di comunicazione seriale creato dalla Modicon nel 1989. Il protocollo modbus definisce il formato e le modalità di comunicazione tra un PLC denominato MASTER ed uno o più PLC denominati SLAVE. Logicamente si può avere un PLC master connesso fino a 247 PLC slave. E’ importante però notare, fin da subito, che è soltanto il PLC master che può iniziare una “conversazione”, ossia una trasmissione di dati verso i PLC slave. Nel protocollo Modbus gli indirizzi partono tutti da 0000 (diversamente dal protocollo Jbus, un protocollo simile al Modbus ma dove gli indirizi partono da 0001). La struttura di un messaggio Modbus è il seguente:

CRC 16

dati

Codice funzione

Indirizzo slave

 Figura 1

Se l’indirizzo slave presenta come valore O, vuol dire che il messaggio è di tipo broadcast. Il formato dei caratteri utilizzato è: 8, N, 1. Pertanto:

 8 bit di dati

 Nessun bit di controllo

 1 bit di stop

Nel modbus esistono fondamentalmente due tipi di errori:

1. **Errori di trasmissione**, ossia errori presenti nel messaggio
2. **Errori operativi**, ossia errori nella funzione da eseguire

Il modbus può essere di tre tipi:

1. **Modbus RTU**
2. **Modbus ASCII**
3. **Modbus TCP**

Se i nodi sono distribuiti su rete LAN si userà chiaramente il modbus TCP. Il Modbus TCP è il Modbus che “gira” su Ethernet. In B&R Automation Studio, il software utilizzato per programmare i PLC B&R, per poter operare con il Modbus si utilizza una libreria particolare denominata **DRV\_mbus**. Tali libreria contiene tutte quelle funzioni atte ad effettuare e gestire una connessione in Modbus. Tale libreria dipende da:

1. **dvFrame**, che è un driver utilizzato per scambiare i dati con determinati dispositivi esterni.
2. **DataObj**, che permette di creare, leggere, scrivere oggetti dinamicamente.

I comandi Modbus vengono eseguiti nella sequenza con la quale essi vengono inseriti nella DATA OBJECT.

E’ bene ricordare un po’ di terminologia:

1. **Coils**, che sono gli I/O digitali che possono essere sia letti che scritti.
2. **Discrete\_input**, che sono gli input digitali che possono essere soltanto letti.
3. **Holding\_register**, che sono registri I/O analogici a 16 bit che possono essere sia scritti che letti.
4. **Input\_register,** che sono input a 16 bit in sola lettura.