Protocol Description

Overview

The MODBUS protocol is a widely used communication standard in industrial automation, facilitating master-slave data exchange. It is known for its simplicity and flexibility, making it suitable for a range of applications in connecting intelligent devices.

In MODBUS communication, messages sent from the master to a slave device contain key components: the slave's address, a command (such as "read register" or "write register"), the associated data, and an error-checking field (using LRC or CRC for checksum). As a messaging protocol, MODBUS is independent of the underlying physical layer, meaning it can be implemented over various mediums like RS485 or USB-to-serial interfaces.

USB interface

ADI-524 has a built-in USB to serial converter, you can plug USB cable and start device configuration as there is no need to install any additional drivers on most operating systems.

However, if you find that you need to install drivers, you can download driver - .V1.00 driver.zip

The ADI-524 allows operation via only one interface at a time. Either RS485 or USB. DIP switch settings are necessary when using both interfaces

MODBUS RTU Framing

The ADI-524 module supports standard RTU Modbus protocol for getting the input status and setting/getting the relays status. RTU Modbus Protocol frame is illustrated below:

Start	Address	Function	Data	RC Check	End
≥ 3.5 Char	4 Bits	8 Bits	N*8 Bits	16 Bits	≥ 3.5 Char

1. **Start Address**: This identifies the specific device or slave on the network. Each device has a unique address to differentiate it from others.
2. **Function Code**: Specifies the type of operation being requested. Common function codes include:
   - 0x04: Read Input Register
   - 0x06: Write Single Register
   - 0x10: Write Multiple Registers
3. **Starting Address**: The address of the first register from which data will be read or to which data will be written

4. **Number of Registers**: Indicates how many registers should be read or written, starting from the specified address.
5. **CRC (Cyclic Redundancy Check)**: This is a checksum used to verify the integrity of the data. It is calculated based on the previous fields and helps detect errors in the transmitted data.

Each part of the request plays an important role in ensuring the correctness of Modbus communication. The start address and function code define the operation, and the CRC checksum protects the accuracy of the message.

MODBUS Command examples

04 function code for reading value from input.
Modbus dear query message

Command: read analog input 1		
Byte	**Byte description**	**Example (HEX)**
0	Slave ID	01
1	Function code	04
2	Starting address High	00
3	Starting address Low	00
4	Number of Registers High	00
5	Number of Registers Low	02
6	CRC AB	71
7	CRC CD	CB

Start	Slave address XX	Function code 04	Starting Number				CRC ABCD 71 CB	End
			High	Low	High	Low		
			00	00	00	02		

Modbus response

Response: Analog Input - 13.40mA		
Byte	**Byte description**	**Example (HEX)**
0	Slave ID	01
1	Function code	04
2	Byte Count	04

Response: Analog Input - 13.40mA		
Byte	**Byte description**	**Example (HEX)**
3	Data 1	41
4	Data 2	6C
5	Data 3	CC
6	Data 4	CD
7	CRC AB	BB
8	CRC CD	30

The number in 32-bit floating point HEX format 0x416CCCCD in decimal is: 13.40

Start	Slave address XX	Function code 04	Number of bytes	Data				Error check	End
				First register		Second register			
				Data 1	Data 2	Data 3	Data 4		

Revision #8
Created 16 February 2025 15:48:22 by Yogesh
Updated 19 March 2025 13:19:59 by Yogesh