

# Bevezetés a Modbus kommunikációba

Modbus szervezet

Teljesen ingyenes, nyílt-forrású rendszer nem licenc köteles

<http://www.modbus.org>

Modbus eszköz kereső motor

<http://www.modbus.org/devices.php>

- soros kommunikációs protokoll
- Modicon (most Schneider Electric) 1979-ben jegyeztette be programozható logikai vezérlők számára (PLC-k).

Egyszerű és robusztus rendszer, ezért alap kommunikációs protokollá nőtte ki magát az iparban

Terjedésének fő okai

- Az ipari alkalmazásokat szem előtt tartva fejlesztették ki
- Nyilvánosan elérhető és jogdíj mentes
- Könnyű telepíteni és karbantartani
- Nem korlátozzák a kereskedőket a felhasználásban

Két típusa létezik a Modbus/ASCII és a Modbus/RTU

## **RTU -**

Remote Terminal Unit: minden nyolc bites bájt tartalmaz két négy bites kódolt hexadecimális értéket. Sajátossága a másik módhoz képest, hogy gyorsabb, illetve minden üzenet folyamatos folyamként kerül elküldésre.

- Kliens – server, azaz master - -slave kapcsolat
- láncba vannak fűzve, max távolság 1200 – 2500m lehet, max 255 Slave fűzhető fel egy buszra, az első elem és az utolsó elem 120ohm-mal le van zárva //specifikáció
- Csillag topológia tilos

- a kommunikáció Request/reply lekérdezés/válasz típusú a Master írja olvassa a Slave-k adatait, a Slave-k figyelik a mestert
- csak egy Master lehet a buszon

## A kapcsolat média lehet

- RS232: max 2 eszköz //kétpont kapcsolat, max hossz 15m
- RS485: Multipont kapcsolat master-slave kapcsolat, 1200 – 2500m max 255 Slave //9600bps
- 1200m fordított arányosság
- TCP/IP korlátlan eszköz szám

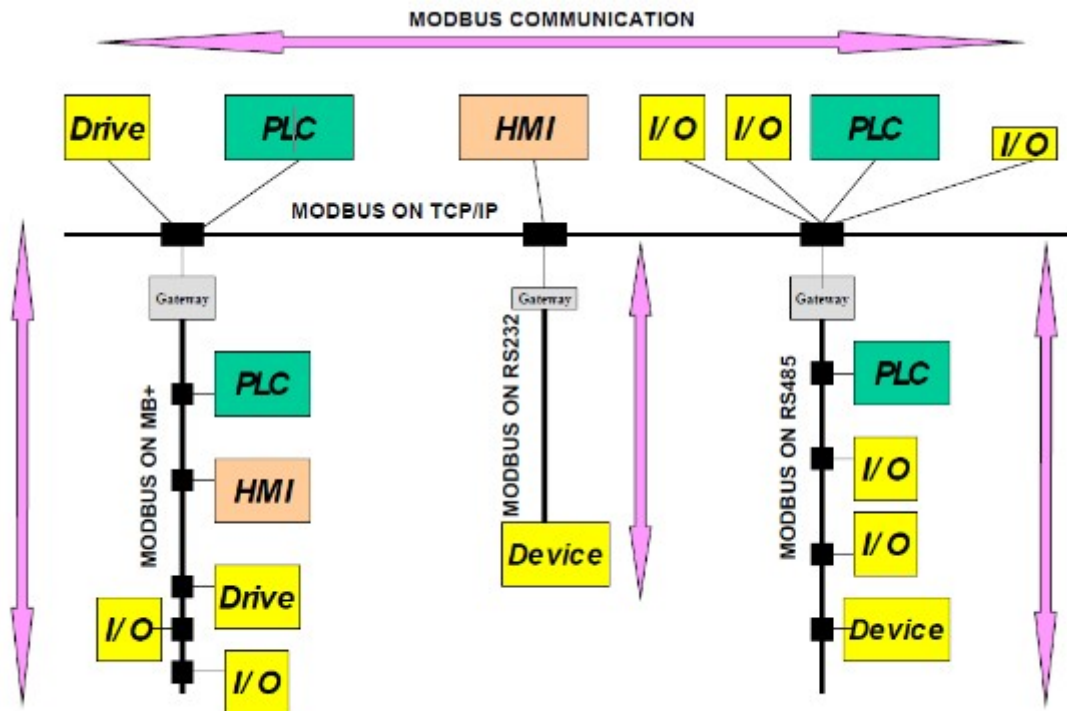


Figure 2: Example of MODBUS Network Architecture

## RS485 topológia:



## A Modbus Protokoll

A Modbus az ipar számára a hetvenes években a Modicon cég által kifejlesztett teljesen nyílt forráskódú protokoll. Fontosabb jellemzői:

- adatcsere típusa: master/slave,
- formátum: 1 start bit + 8 adatbit + paritás + 1 vagy két stop bit,
- előre definiált műveleti kódok (olvasás és írás regiszterekbe),
- maximális frame méret: 246 adatbyte,
- maximum 255 cím

A soros Modbus kapcsolatnak két alaptípusú átviteli módja van: ASCII és RTU (= remote terminal unit). Az átviteli mód soros kommunikációban azt az utat határozza meg, ahogyan a Modbus üzenetek kódolva vannak. Modbus/ASCII-vel az üzenetek ASCII formátumban vannak, míg a Modbus/RTU formátum bináris kódolást használ, így az üzenetek nem olvashatók az ellenőrzés során, de a file méretük jelentősen csökken, így sokkal nagyobb mennyiségű adatot képes továbbítani azonos időtartam alatt. Egy Modbus valamennyi eleme ugyanazt a soros átviteli módot használja.

Ahhoz hogy arduino környezetben tudjuk használni az itt felsoroltakat, a következőkre lesz szükségünk:

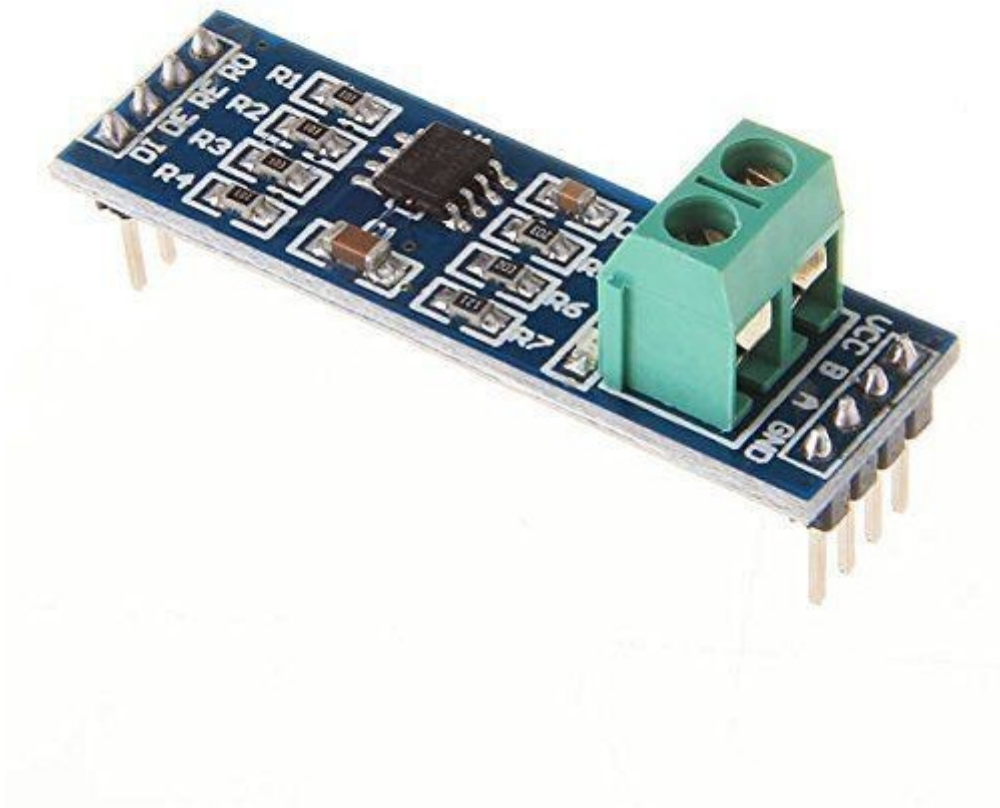
Hardver komponens:

- maxim MAX485

példa kapcsolat:

adatlap

vagy:



<http://www.ebay.com/itm/5PCS-MAX485-RS-485-Module-TTL-to-RS-485-module-for-Arduino-Raspberry-pi-/381374599127?hash=item58cbb28fd7:g:AZMAAOSwoQ1TmaZv>

és software komponensekre:

arduinoRS485-Modbus-library-v0-4.zip letölthető az alábbi web-címről is

Példa Arduino-kód: //forrás [www.cooking-hacks.com](http://www.cooking-hacks.com)

```
/*
 * RS-485 Module
 *
 * Copyright (C) Libelium Comunicaciones Distribuidas S.L.
 * http://www.libelium.com
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 * a
```

```
* This program is distributed in the hope that it will be useful,  
* but WITHOUT ANY WARRANTY; without even the implied warranty of  
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
* GNU General Public License for more details.  
*  
* You should have received a copy of the GNU General Public License  
* along with this program. If not, see http://www.gnu.org/licenses/.  
*  
* Version:          1.0  
* Design:           David Gascón  
* Implementation:   Ahmad Saad  
*/  
  
// Include these libraries for using the RS-485 and Modbus functions  
#include <RS485.h>  
#include <ModbusMaster485.h>  
#include <SPI.h>  
  
// Instantiate ModbusMaster object as slave ID 1  
ModbusMaster485 node(254);  
  
// Define one address for reading  
#define address 101  
  
// Define the number of bytes to read  
#define bytesQty 2  
  
void setup()  
{  
  
    // XBee in SOCKET0  
    pinMode(5, OUTPUT);  
    digitalWrite(5, LOW);  
  
    // Power on the USB for viewing data in the serial monitor  
    Serial.begin(115200);  
    delay(100);  
    // Initialize Modbus communication baud rate  
    node.begin(19200);  
  
    // Print hello message
```

```

Serial.println("Modbus communication over RS-485");
delay(100);
}

void loop()
{
  // This variable will store the result of the communication
  // result = 0 : no errors
  // result = 1 : error occurred
  int result = node.readHoldingRegisters(address, bytesQty);

  if (result != 0) {
    // If no response from the slave, print an error message
    Serial.println("Communication error");
    delay(1000);
  }
  else {

    // If all OK
    Serial.print("Read value : ");

    // Print the read data from the slave
    Serial.print(node.getResponseBuffer(0));
    delay(1000);
  }

  Serial.print("\n");
  delay(2000);

  // Clear the response buffer
  node.clearResponseBuffer();
}

```

PC-s környezetbe

A modbus.org rengeteg hasznos programot, és egyéb library-kat dokumentációt tartalmaz. Ehhez mindössze egy RS232 – RS485 átalakítóra van szükség. A tesztelni kívánt eszközön kívül persze. :) USB változat:



<http://www.ebay.com/itm/1PCS-CH340-USB-to-RS485-485-Converter-Adapter-Module-For-Win7-Linux-XP-Vista-/151911412109?hash=item235ea0318d:g:WcoAAOSwcBhWaVK1>

vagy lehet klasszikus RS232 – RS485 átalakító is:



<http://www.ebay.com/itm/RS-232-RS232-to-RS-485-RS485-Serial-Adapter-Converter-TOP-/322040261339?hash=item4afb186adb:g:OX0AAOSwcUBYEciQ>

a továbbiakban ezt a modult mutatom be.:

# RS485 Relay

