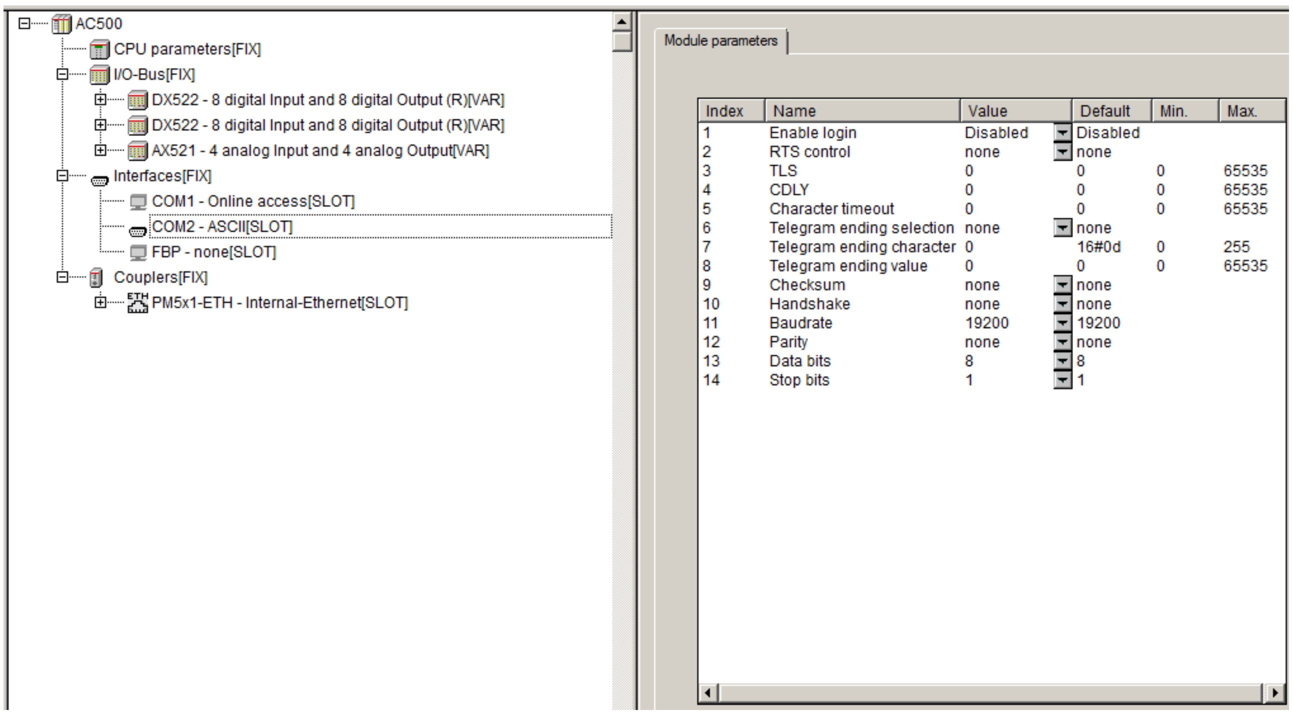
31343 Introduction to Programmable logic Controllers
Exercise 8 : “ex8_serial”

Purpose:
In this exercise you will communicate with an external system via serial communication (RS232) from the PLC. Many sensors and systems transfer data through serial communication and usually using a specialised protocol. This means that it is necessary to make a program in the PLC that is able to interpret this protocol.
In this exercise the serial communication will be run in ASCII mode which is the mode to communicate with equipment using a non-standard protocol.

Setup:
The idea is to communicate with a building management system, which manages light and indoor climate of a building. A simplified version of this is simulated by an Arduino Uno micro controller board. This system simulates one room with a light, a heater and a temperature sensor.

This management system communicates serially with 19200 bit/s, 8 data bits, 1 stop bit and no parity or flow control.
This is setup in the “PLC Configuration” tab by folding out the “Interfaces[FIX]” part. Since we are using the communication port “COM2” right click on this and choose “Replace Element”. Choose “COM2 – ASCII” from the menu, and verify that the settings are correct (compare with the screen-shot below).



```
AC500
    CPU parameters[FIX]
    I/O-Bus[FIX]
        DX522 - 8 digital Input and 8 digital Output (R)[VAR]
        DX522 - 8 digital Input and 8 digital Output (R)[VAR]
        AX521 - 4 analog Input and 4 analog Output[VAR]
    Interfaces[FIX]
        COM1 - Online access[SLOT]
        COM2 - ASCII[SLOT]
        FBP - none[SLOT]
    Couplers[FIX]
        PM5x1-ETH - Internal-Ethernet[SLOT]
```

Module parameters

Index	Name	Value	Default	Min.	Max.
1	Enable login	Disabled	Disabled		
2	RTS control	none	none		
3	TLS	0	0	0	65535
4	CDLY	0	0	0	65535
5	Character timeout	0	0	0	65535
6	Telegram ending selection	none	none		
7	Telegram ending character	0	16#0d	0	255
8	Telegram ending value	0	0	0	65535
9	Checksum	none	none		
10	Handshake	none	none		
11	Baudrate	19200	19200		
12	Parity	none	none		
13	Data bits	8	8		
14	Stop bits	1	1		

The Arduino is able to simulate a number of different values from the building management system. It uses the following protocol and data is only send when asked for.

ASCII Command	Response	Meaning
V	v2.5	Inquire version of system (for testing connection)
L	l%d	Status of light - %d = 0: Light off - %d = 1: Light on
H	h%d	Status of heater - %d = 0: Heater off - %d = 1: Heater on
T	t%f	Returns room temperature - %f: Temperature in °C. Fx: t20 : Means 20°C.
A	ok	Turn on light
B	ok	Turn off light
C	ok	Turn on heater
D	ok	Turn off heater

All commands are expected to be terminated by carriage return and line feed. All responses are also terminated by carriage return and line feed.

Part 1:
Connect the Arduino board to COM2 on the PLC. Create a program that ask the Arduino for its version number and stores the response in a local variable. Remember to power up the Arduino by plugging in its USB cable to the computer.
Hint: Use the COM_REC and COM_SEND functions to communicate using the serial port.

Part 2:
Create a program that makes switch one on the PLC control box control the light in the room. Add to this program a functionality that continuously (max. every second) polls the Arduino for the state of the light and displays this using the green light on the control box.

Part 3:
Add to the program of part 2 a system that keeps the rooms temperature between 20°C and 22°C. Turn the heater off and on to achieve this.

Part 4:
Enhance the program from part 3 that keeps the temperature between 20°C and 22°C to store the current temperature in a value using Binary Coded Decimal.
Also calculate the rate of change in °C/second.

Part 5:
Setup a trace in CoDeSys that displays the temperature and the temperature rate over a 1 minute period.

Part 6:
Try to create a program that controls the temperature to be exactly 21°C. Comment on how the temperature control system could be improved with respect to the time at takes to reach the set point (21°C in this case).

Hand In:
Hand in a commented screen-shot displaying the traces from part 5 and the project file (<name>.pro) from part 4 and 6, together with a small written description of the functionality of your solutions.