

# Advant Controller 31

## Intelligent Decentralized Automation System

Example program for 907 AC 1131  
Output, reception and comparison of  
ASCII characters via serial interface  
(DRUCK/EMAS replacement)

<b>1. Which components are required?</b> .....	<b>2</b>
<b>2. Possible configurations</b> .....	<b>3</b>
2.1. Serial communication with one basic unit .....	3
2.2. Serial communication with two basic units .....	4
<b>3. Short description for commissioning</b> .....	<b>5</b>
3.1. Configuration 1 (one basic unit) .....	5
3.2. Configuration 2 (two basic units).....	5
<b>4. The following steps have to be carried out</b> .....	<b>6</b>
4.1. Connecting the programming cable to the basic unit and the PC .....	6
4.2. Supplying the basic unit with 24 V DC (power supply unit) and switching it on.....	6
4.3. Copying the example program for the basic unit.....	6
4.4. Connecting the serial cable .....	7
<b>5. Functional description</b> .....	<b>8</b>
5.1. BSP_ASCII_SND_REC_CMP.....	8
5.1. BSP_ASCII_ECHO.....	8

**ABB STOTZ KONTAKT**



# 1. Which components are required?

Hardware components:

- 1 x basic unit (07 KT 95, 07 KT 96, **07 KT 97**, 07 KT 98)  
*or*  
2 x basic unit (07 KT 95, 07 KT 96, **07 KT 97**, 07 KT 98)
- 1 x power supply unit 24 V DC (e.g. **07 NG 34**)
- 1 x programming cable for the basic unit (**07 SK 90**)  
*or*  
1 x ARCNET cable incl. two terminating resistors
- 1 x serial cable between the basic units

Software:

- 1 x programming software for the basic unit 907 AC 1131
- 1 x example program for the basic unit (BSP\_ASCII\_SND\_REC\_CMP.PRO)  
*or*  
1 x example program for the basic unit (BSP\_ASCII\_SND\_REC\_CMP.PRO) and  
1 x example program for the basic unit (BSP\_ASCII\_ECHO.PRO)

Miscellaneous equipment for programming:

- 1 x PC (see 907 AC 1131 part 2, programming and block library, installation)
- 1 x documentation:
  - 907 AC 1131 part 1 (hardware and system technology)
  - 907 AC 1131 part 2 (programming and libraries)

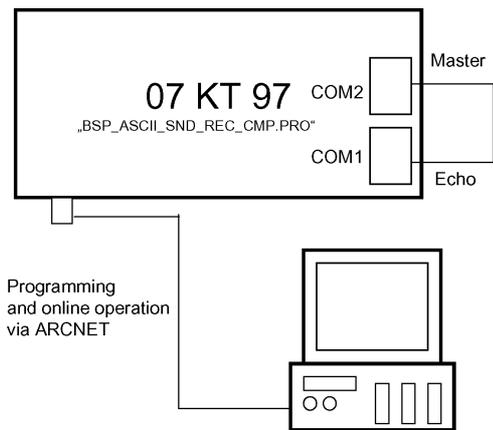
In the following description you will find several references to the documentation.

## 2. Possible configurations

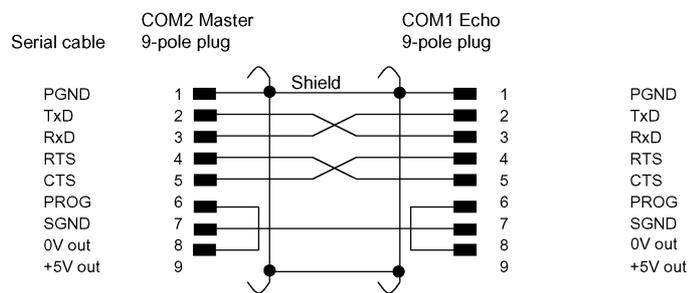
### 2.1. Serial communication with one basic unit

In this configuration both serial interfaces of one basic unit are operated simultaneously as communication interfaces. While the application program is running, no free serial interface is available for online operation. In this case the online operation is only possible via ARCNET. If this possibility is not available, select the second configuration possibility (see section 2.2).

For configuration 1 you only need one of the above mentioned basic units and the example program "ASCII\_SND\_REC\_CMP.PRO". In this example program, COM2 of the basic unit takes on the master function (sending, receiving and comparing) and COM1 responds to the telegrams (ECHO).



Configuration with one PLC

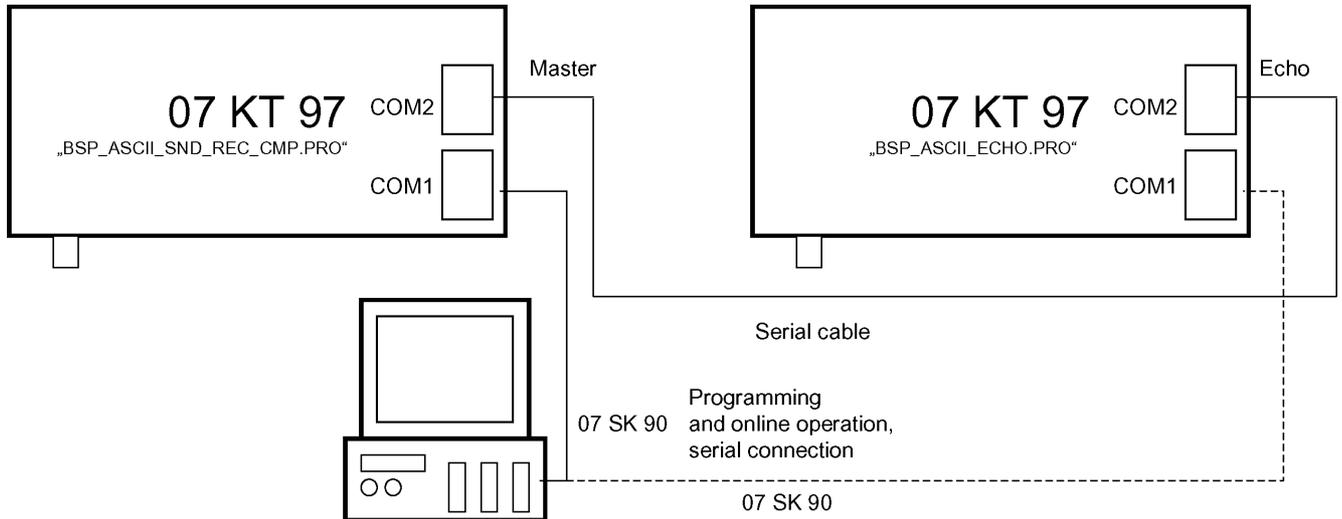


Serial cable between the interfaces

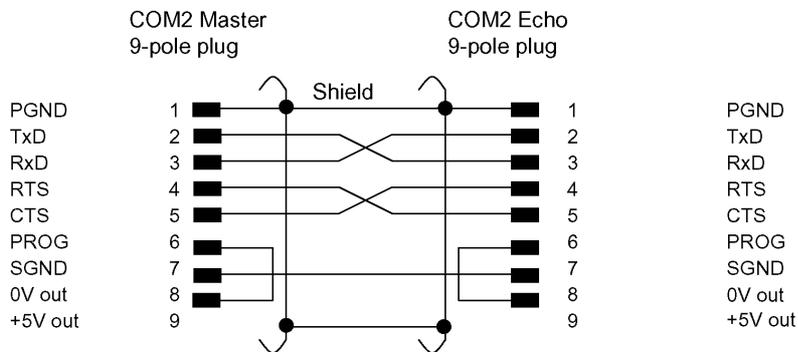
## 2.2. Serial communication with two basic units

In this configuration, one PLC takes on the master function (sending, receiving and comparing) and the second PLC responds to the telegrams (ECHO). At both PLCs COM2 is initialized as communication interface which allows online operation via COM1 at any time. It is also possible to establish the online operation via the ARCNET connection as described for the first method.

For configuration 2 you need two devices of the above mentioned basic unit and the example programs "BSP\_ASCII\_SND\_REC\_CMP.PRO" and "BSP\_ASCII\_ECHO.PRO".



### Configuration with two PLCs



Serial cable between the interfaces

### **3. Short description for commissioning**

#### **3.1. Configuration 1 (one basic unit)**

1. Connect the PLC to 24 V DC.
2. Copy and start the example program "BSP\_ASCII\_SND\_REC\_CMP.PRO" using the programming software 907 AC 1131.
3. Connect the serial cable to COM1 and COM2 of the basic unit.
4. Test the program.

#### **3.2. Configuration 2 (two basic units)**

1. Connect the PLC to 24 V DC.
2. Copy the example program "BSP\_ASCII\_SND\_REC\_CMP.PRO" to the first basic unit and the example program "BSP\_ASCII\_ECHO.PRO" to the second basic unit using the programming software 907 AC 1131 and start them.
3. Connect the serial cable to COM2 of the first basic unit and COM2 of the second basic unit.
4. Test the program.

## 4. The following steps have to be carried out

### 4.1. Connecting the programming cable to the basic unit and the PC

There are two possibilities for programming. You can either perform programming via a serial connection or via an ARCNET connection.

#### *Serial connection:*

The programming cable 07 SK 90 connects the PLC to the PC. The 9-pole plug of this cable has to be connected to COM1 of the 07 KT 9x. The other cable end has to be connected to the PC's serial interface (e.g. COM1). (See documentation 907 AC 1131 part 1, interface cable 07 SK 90.)

#### *ARCNET connection:*

First, the node addresses of the PLC and PC have to be set. The basic unit is designated for example as NODE number 1. For this, set DIL switch 8 to ON; all other switches remain in the OFF position. The ARCNET card now has to be installed in the PC (see documentation 907 AC 1131 part 2, Installing the ARCNET card). For the connection between the PLC and the PC use a corresponding coaxial cable with a characteristic impedance of 93Ω. At both cable ends a terminating resistor of 93Ω has to be installed (refer to documentation 907 AC 1131 part 1, ARCNET descriptions). Connect this cable to the corresponding connectors.

### 4.2. Supplying the basic unit with 24 V DC (power supply unit) and switching it on

Now the devices have to be supplied by 24 V DC.

**Basic unit:** At the basic unit 0 V has to be connected to terminal (M) and 24 V DC to terminal (L+). (See documentation 907 AC 1131 part 1, basic units, connection of supply voltage.)

### 4.3. Copying the example program for the basic unit

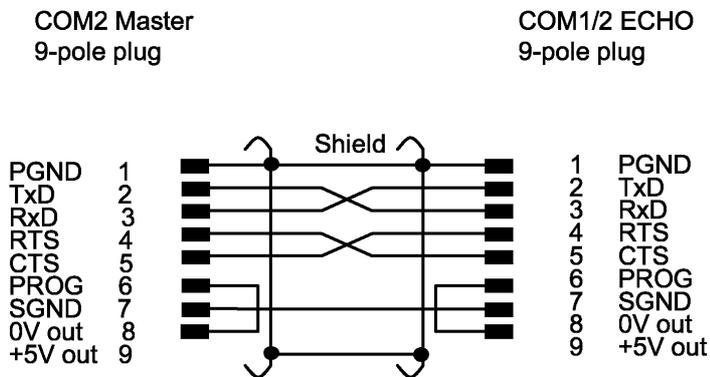
Before copying the example program you have to make sure that the PLC is supplied with voltage, that the programming cable is connected and that the PC is running. First, start the 907 AC 1131 programming software on the PC. Then, continue as follows:

- Open the example program "*BSP\_ASCII\_SND\_REC\_CMP*" by selecting the menu item *<File>* *<Open>*.
- Prior to sending the example program to the basic unit, it has to be compiled. For that purpose press *<F11>*.
- Now the communication parameters have to be installed. Using the menu item *<Online>* *<Communication parameters>* the corresponding interface parameters can be set. Now define whether the communication with the PLC is done using a serial connection or an ARCNET connection. In our example the driver ABB RS232 was installed on *COM1 (19200 baud, no parity, 1 stop bit)*.
- In the menu *<Online>* *<Login>* you have to activate the sending process. Confirm the information about the existing program stored on the PLC with *OK*.
- Now you are asked "The program was changed! Do you want to load the new program?". Answer this question with *Yes* to load the entire program.
- Finally, start the program by pressing *<F5>*.  
The program is running.
- If you want to use configuration 2 (with two basic units), repeat all steps and load the example program "*BSP\_ASCII\_ECHO*" to the second PLC.

(See documentation 907 AC 1131 part 2, software description, project management and general online functions.)

#### 4.4. Connecting the serial cable

The serial cable has to be prepared according to the following description.



#### Serial cable between the interfaces

For configuration 1 (one basic unit) the cable connects COM2 (master) and COM1 (Echo) of the basic unit. If configuration 2 is used (two basic units), the cable is connected between COM2 (master) of the first basic unit and COM2 (Echo) of the second basic unit.

## 5. Functional description

The example program serves for sending, receiving and comparing ASCII characters via a serial connection. For this purpose, two programs were created. The first program (BSP\_ASCII\_SND\_REC\_CMP) has to control the telegram traffic. This program sends a telegram via the serial interface. The telegram is received by the echo program (BSP\_ASCII\_ECHO) which returns it to the master program again. The master program then evaluates whether the telegrams are the same.

### 5.1. BSP\_ASCII\_SND\_REC\_CMP

This program contains the complete functionality of the master program and the echo program. It can run completely on one basic unit (configuration 1, one basic unit), but it can also be used as master program for configuration 2 (two basic units).

COM2 of the basic unit is initialized as *master interface* via the COMINIT block. A successful initialization of the interface is indicated by steady light at output A63,7. If this output is flashing, the communication is disturbed (e.g. since the cable is not connected to COM2 anymore). The subroutines PR1 and PR2 are alternately called via a step chain. This program change is performed time-dependent. Each time a program was active for at least 5 seconds the program is changed after the next successfully received telegram.

The subroutine PR1\_einf\_Text\_SND\_REC\_CMP sends the text "I am a simple text." to COM2. Then, the response telegram is received and finally compared with the sent telegram. If the comparison is successful, the ready message is output and the procedure is started again. If an error occurs, a corresponding error message is output and the communication is initiated anew. All incoming telegrams are counted and stored in a circular buffer. The counter value is displayed at the outputs A62,0..A62,7. If no new telegram is received the counter stops.

The subroutine PR2\_Text\_mit\_Var\_SND\_REC\_CMP appends the variable Zahl to the text "I am a text containing a variable: " and then sends this combination to COM2. After this, a response telegram is received. The variable is extracted from the response telegram and finally the response telegram is compared with the sent telegram. If the comparison is successful, the ready message is output and the procedure is started once again. If an error occurs, a corresponding error message is output and the communication is initiated anew. All incoming telegrams are counted and stored in a circular buffer. The counter value is displayed at the outputs A62,8..A62,15. If no new telegram is received the counter stops.

COM1 of the basic unit is initialized as *ECHO interface* via the COMINIT block. The successful initialization of the interface is indicated by steady light at output A63,6. If this output is flashing, the communication is disturbed (e.g. since the cable is not connected to COM1 anymore).

*NOTE:* For configuration 2 (two basic units) this indication is ignored because the echo program on the other PLC is used.

The communication is controlled by the master program. The master program sends telegrams to the echo program and the echo program returns the same telegrams as response telegrams. All incoming telegrams are counted and stored in a circular buffer.

### 5.2. BSP\_ASCII\_ECHO

This program contains the echo program. It is intended for configuration 2 (two basic units). The program runs only in combination with a master PLC.

COM2 of the basic unit is initialized as *ECHO interface* via the COMINIT block. A successful initialization of the interface is indicated by steady light at output A63,7. If this output is flashing, the communication is disturbed (e.g. since the cable is not connected to COM2 anymore).

The communication is controlled by the master program. The master program sends telegrams to the echo program and the echo program returns the same telegrams as response telegrams. All incoming telegrams are counted and stored in a circular buffer. The counter value is displayed at the outputs A62,0..A62,15. If no new telegram is received the counter stops.