

```

type
    TCP_Buffer : array [1..522] of byte;
end_type

var_output
    SID:          BYTE          (* ARCNET node ID of sender *)
    DID:          BYTE          (* ARCNET node ID of recipient (destination) *)
    Data_Len:     WORD           (* Size of payload data in bytes *)
    Data_Array:   TCP_Buffer    (* Byte array containing payload data *)
    Received_Data: BOOL         (* Flag indicating the integrity of received data *)
end_var

var_input
    Buffer:        TCP_Buffer    (* Buffer handed over from FB_INIT_SEND_RECEIVE *)
end_var

var
    I:            WORD          (* internal counters *)
    K:            WORD          (* internal counters *)
    Length:       WORD          (* Total size of data in Buffer in bytes *)
    Error_Code_Write: DWORD      (* Error code for debugging *)
    Error_Code_Read:  DWORD      (* Error code for debugging *)
end_var

(* FB extracting transmitted payload data out of Buffer into Data_Array *)
(* Other information being stored in the respective variables *)

Length := byte_to_word(Buffer[1]) + (* Total size of transmitted data package
in bytes *)
        byte_to_word(Buffer[2]) * word#256;

if Length >= word#8 and Buffer[3] = byte#83 then (* sufficient size and SOHARD protocol
signature 'S' *)

    if Buffer[4] = Byte#4 then (* OP_READ_DATA *)

        Data_Len := byte_to_word(Buffer[5]) + (* Size of payload data in bytes *)
                    byte_to_word(Buffer[6]) * word#256
                    - word#6;

        SID := Buffer[7]; (* Sender node ID *)
        DID := Buffer[8]; (* Destination node ID *)

        for I := word#1 to Data_Len by word#1 do (* take-over of Buffer into Data_Array *)
            K := I+word#12;
            Data_Array[I] := Buffer[K];
        end_for;

        Received_Data := true; (* Flag indicating reception of payload data *)

    else
        if Buffer[4] = byte#3 then (* OP_READ_ERROR *)

            Error_Code_Read := byte_to_dword(Buffer[7]) + (* Extracting error code for
optional debugging *)

```

```
        byte_to_dword(Buffer[8]) * dword#16#100 +
        byte_to_dword(Buffer[9]) * dword#16#10000 +
        byte_to_dword(Buffer[10]) * dword#16#1000000;

    Received_Data := false; (* no payload data received *)

elseif Buffer[4] = byte#6 then (* OP_WRITE_RESULT *)

    Error_Code_Write := byte_to_dword(Buffer[7]) + (* Extracting error code for
optional debugging *)
        byte_to_dword(Buffer[8]) * dword#16#100 +
        byte_to_dword(Buffer[9]) * dword#16#10000 +
        byte_to_dword(Buffer[10]) * dword#16#1000000;

    Received_Data := false; (* no payload data received *)
else
    Received_Data := false; (* no payload data received *)
end_if;
end_if;

else
    Received_Data := false; (* Package not according to SOHARD protocol *)
end_if;
```