(* In this example, the bar code data will vary in length. Therefore, the method for receiving characters is as follows:
1. Watch the RX Count Register (%R100).
2. When %R100 is nonzero but unchanged for 250mS, assume the transmission from the bar code reader is complete.
3. When the receive is complete, copy the data into another bank of registers (%R201-%R210), clear the receive buffer (%R101-110) and reset the Receive block.
4. The received characters stored in %R201-210 are displayed on Screen #1 on the OCS. *)

Rung 1

ALW_ON
%S007

```
OPEN
MJ1/Com Option ─ PORT
         9600 ─ Baud
         None ─ Parity
            8 ─ Data Bits
            1 ─ Stop Bits
         None ─ Handshake
      Generic ─ Protocol
```

```
( ) Force Screen: 1
     Barcode
XXXXXXXXXXXXXXXXXXX
Length, with CR
     #####
```

(* Characters are received from a bar code reader. When not reset, the Receive Block is enabled, allowing characters to be placed in the Receive buffer starting at register %R101 [Rx_Buf]. The buffer length is 254 characters (one less than the maximum allowed), or 127 words. Note that the [RX Count] parameter is %R100, located one word register before the buffer. So the length of the characters received is contiguous with the characters received. *)

Rung 2

RX_Reset
%T00003

```
RECV
MJ1/Com Option ─ PORT
           254 ─ Bytes
                           Rx_Buf
                  Data ─ %R00101
                           Rx_Cnt
              RX Count ─ %R00100
```

(* When not reset, check to see that characters have been received, but RX_Cnt is not changing; this is assumed to be an indication that transmission by the bar code reader is complete. To do this, compare RX_Cnt with RX_Cnt_Prv, which is the value of RX_Cnt from the previous scan. Also check RX_Cnt to see that it is greater than zero, indicating characters have been received. Also, since Rx_Cnt is greater than zero it cannot be -1, which would indicate that the port is not open.

If those conditions are all true, trigger the timer; if the timer completes, assign a 1 to Rx_Cmplt, indicating the bar code read into the buffer so far is a complete barcode. *)

Rung 3

RX_Reset
%T00003

```
EQ_INT
      Rx_Cnt
%R00100 ─ IN1
  Rx_Cnt_Prv
%R00012 ─ IN2
```

```
GT_INT
      Rx_Cnt
%R00100 ─ IN1
      0 ─ IN2
```

```
TON  %R00001
     0.01s
     25 ─ PT
```

Rx_Cmplt
( )
%T00002

(* When not reset, if the current received count of characters is different than the value from the previous scan,, then copy the former to the the latter, for comparison in the previous rung on the next scan. *)

Rung 4

RX_Reset
%T00003

```
NE_INT
      Rx_Cnt
%R00100 ─ IN1
  Rx_Cnt_Prv
%R00012 ─ IN2
```

```
MOV
word
      Rx_Cnt
%R00100 ─ IN
              Rx_Cnt_Prv
           Q ─ %R00012
```

(* When the Receive is complete, cp[u the count of the received characters and the receive buffer characters to another register buffer, then clear the received buffer count to 0 along with the receive buffer (128 words = 1 word + 254 characters). Then assign a 1 to RX_Reset bit, which makes the next ***ONE*** scan a reset scan, which removes power from the Receive Block; the scan after that, the RECV instruction will start a new read. *)

Rung 5

Rx_Cmplt
%T00002

```
BMV
word
      Rx_Cnt
%R00100 ─ IN
              Rx_Disp_cnt
           Q ─ %R00300
      128 ─ N
```

```
Fill
word
      0 ─ IN
              Rx_Cnt
           Q ─ %R00100
    128 ─ N
```

RX_Reset
( )
%T00003