(\* In this example, the bar code data will vary in length. Therefore, the method for receving characters is as follows:

In this Stample, use par code data will very in angut. Interesting, use menors for recenting standards is because. I watch the RX Count Register (RX100). 2. When KY receive is complete, copy the data into another bank of registers (RX201-KR210), clear the receive buffer (%R101-110) and reset the Receive block. 8. The received characters stored in %R201-210 are displayed on Screen #1 on the OCS.\*) OPEN () Force Screen: 1 N ON ¹⊣ī⊦ Barcode MJ1/Com Option-PORT 9600-Baud \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Length, with CR None-Parity \*\*\*\*\* 8-Data Bits 1-Stop Bits None-Handshake Generic-Protocol RS-232-Mode 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. ") ²⊢ī/⊢ MJ1/Com Optio PORT 6T00003 254-Byte Rx\_But Data %R00101 Rx Crr RX Cour %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 the \*\*\*TIMED\*\*\* receive complete bit, indicating the bar code read into the buffer so far is a complete barcode. \*) GT IN Timed\_Rx\_Cmplt EQ INT TON %R00001 0.01s -ī/H Rx\_Cnt %R00100-IN1 %R00100-IN1 6700003 %T00002 Rx\_Cnt\_Prv %R00012-IN2 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. \*) NE INT MOV -ī/ŀ Rx\_Cnt %R00100-Rx\_Crit %R00100-IN T00003 Rx\_Cnt\_Prv Q-%R00012 Rx\_Cnt\_Prv %R00012 (\* When the \*\*\*TIMED\*\*\* receive buffer (128 words = 1 word + 254 characters). Then assign a 1 to RX\_Reset bit to make 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. \*) ned\_Rx\_Cmplt BM\ BM RY Rese 5-1--<del>(</del>)-Rx\_Crit %R00100 Rx\_Crit %R00100-IN 0—IN T00002 %T00003 RX\_Disp\_crit Q=%R00300 Cmp\_Cnt Q-%R00500 Q-%R00100 128-128-128-Calculate the left shift necessary to move the last three characters in the Compare buffer to the start of the Compare buffer, so the first two of those last three will be in the low- and high-byte of %R501, and the last byte will be in the low-byte of %R502 [Cmp\_CR]. Then do the shift, compare 'BC' to those first two bytes, mask out the high-byte of Cmp\_CR to Cmp\_CR to Cmp\_CR to ASCII code 13 (carriage return), and finally assign 1 to Valid\_code if all compares were true ') GT\_INT MULTI SHIFT d\_Rx\_Cmpl SUB Valid\_code\_oneshot CMP STR AND EQ IN -Ō-YTE Cmp\_CR %R00502 Cmp\_Cnt %R00500 Cmp\_Shift %R00499-IN Cmp\_CR %R00502= %T00004 T00002 Cmp\_Shit R00499 Cmp\_Buf %R0050 'BC'-IN1 Cmp\_CR R00502 Cmp\_But %R00501-IN2 255-IN2 128 Cmp\_Shift %R0049 ALW\_ON %S00 Nul\_char %R00498 OUT %R00497 (\* Use Start/Stop pattern for successful bar code bit - Start: valid code oneshot bit from previous rung Stop: part reaches position 2\*) PART POS2 Valid code onesho Code succee -1 F ┥៸ᢆ⊦ -()-%T00004 %10002 %T00005 GRN LIGHT -Ō-%Q0011 Code succeer Retract cylinder -1 1--()-%700005 %T00006 (\* Use Start/Stop pattern for failed bar code bit - Start: oneshot when \*\*\*TIMED\*\*\* receive completes Stop: code succeeded from previous rung \*) Timed Rx Cmpl Code Code fai ٦Ē -1/1--()-%T00002 %T00005 %T00007

Main Loop Logic Block: main

|    | (°  | Code_fail<br>  |               | RED_LIGHT<br> |  |
|----|---|--|---------------|---------------|--|
|    | Unless the cylinder should be retracted, activate the cylinder-exend soloenoid until the extended cylinder prox is reached *) |  |               |               |  |
| 9  | Retract_cylinder<br>////////////////////////////////////  | Cylinder_fully_extend<br>//  | SOL_EXT       |               |  |
|    | When the cylin  | cylinder should be retracted, activate the cylinder-retract solenoid until the retracted cylinder prox is reached *) |               |               |  |
| 10 | Retract_cylinder  | Cylinder_fully_retract   | SOL_RET<br>() |               |  |