G9SP RS-232 Communication to Omron's CJ PLC's

Rev 1.0

Micheal Paradiso

This document will walk you through a step by step setup for communication from a G9SP to a Omron's CJ PLC using Ladder Logic to do Serial Communications.

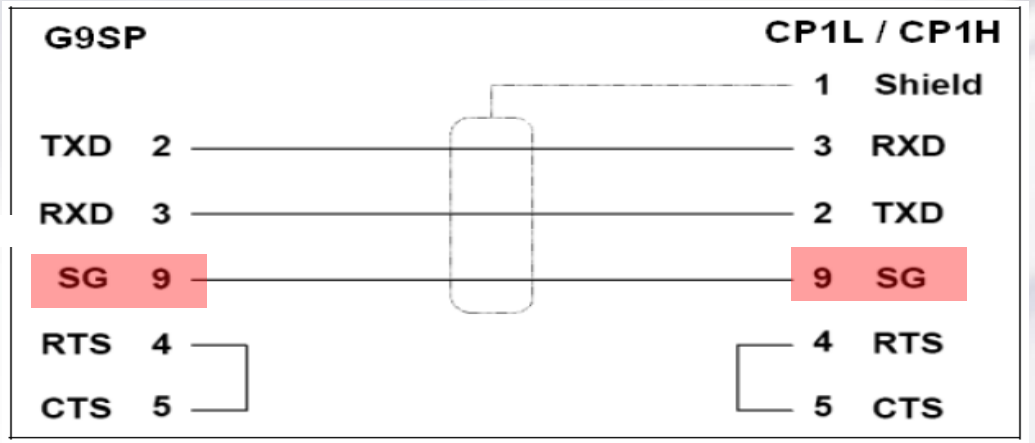
PLC example code is contain in two files, one for CJ1 PLC's and one for CJ2 PLC's. These files can be opened using Omron's CX-Programmer software and down loaded to the PLC.

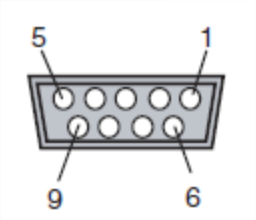
Files

G9SP_RS232_CJ1_Example_Rev1.cxp
G9SP_RS232_CJ2_Example_Rev1.cxp

Cable For G9SP RS-232 Coms

If you are going to make your own cable Omron RS-232 com ports do not follow U.S. standards, all com ports have the signal ground on pin 9 not pin 5 for RS-232 coms. This can cause problems if wired wrong.





Pin	Abbr.	Signal	Signal direction
1	FG	Frame ground	---
2	SD (TXD)	Send data	Outputs
3	RD (RXD)	Receive data	Inputs
4	RS (RTS)	Request to send	Outputs
5	CS (CTS)	Clear to send	Inputs
6	5 V	Power	---
7	DR (DSR)	Data set ready	Inputs
8	ER (DTR)	Data terminal ready	Outputs
9	SG (0 V)	Signal ground	---
Connector hood	FG	Frame ground	---

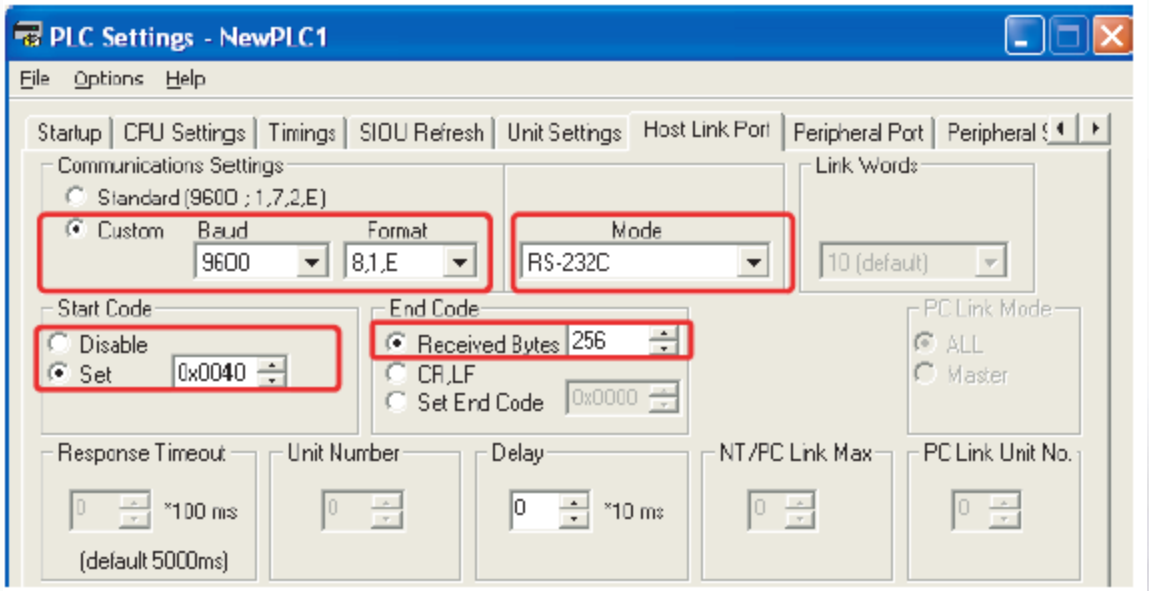
SYSMAC always in control

RS-232 Coms Setup For CJ PLC

For RS 232 communication, not much set up is needed , in the PLC just ensure that the com port setting on your PLC match your G9SP port setting as well as the wiring.
Once the setting are correct all that is needed at this point is to set up the “Command/ Response” protocol messaging in the PLC ladder logic. This format of massaging can be used by any Omron device or third party PLC (AB, Siemens, Mitsubishi, or any PLC) to poll data out of G9SP controller , the same is true with Ethernet communications they both work in the same way for response data.

Set serial port 1 as given below in the PLC Setup using the CX-Programmer.

Parameter	Description	Set value for this example
Baud rate	Serial port baud rate setting	9600, 8, 1, E
Mode	Communications protocol	RS-232C (no-protocol)
Start code	Send/receive data start code	0x0040
End code	Send/receive data end code	Reception bytes: 256



Setting in the G9SP are not required, all that is needed it to make the PLC setting match the G9SP. Here are the needed setting for the PLC.

always in control

Used PLC Memory Locations

This the memory location in the PLC that are used

Start address	End address	Contents
D100	D108	RS-232C send command
D200	D298	RS-232C reception response
D323		Reception size work area
D300	D333	Checksum calculation work area
CIO 200	CIO 201	G9SP communications reception data (written to G9SP-series Controller)

These 8 words are used as the send command set to the G9SP

These 98 words are used as the response data from the G9SP

These words are used for the checksum code

These 2words are the only spot you can write to the G9SP.

PLC Register for G9SP Data

Data point Name	I/O Type	Address	I/OI Description	
G9SP_Input_Com_Bit0_15	CHANNEL	CIO 200	This word writes to 0-15 data Input Com bits in the G9SP	writres to G9SP
G9SP_Input_Com_Bit16_31	CHANNEL	CIO 201	This word writes to 16-31 data Input Com bits in the G9SP	writres to G9SP
G9SP_Output_Com_Bits0_15	CHANNEL	D203	This word reads from 0-15 data ouput com bits in G9SP	Read From G9SP
G9SP_Output_Com_Bits16_31	CHANNEL	D204	This word reads from 16-31 data ouput com bits in G9SP	Read From G9SP
G9SP_Safety_Inputs	CHANNEL	D205	This word reads the Safety input status	Read From G9SP
G9SP_Safety_Outputs	CHANNEL	D208	This word reads the Safety output status	Read From G9SP

+0	Optional Communications Transmission Data	4 bytes
+4	Safety Input Terminal Data Flags	6 bytes
+10	Safety Output Terminal Data Flags	4 bytes
+14	Safety Input Terminal Status Flags	6 bytes
+20	Safety Output Terminal Status Flags	4 bytes
+44	Safety Input Terminal Error Causes	24 bytes
+60	Safety Output Terminal Error Causes	16 bytes
+62	Reserved	2 bytes
+64	Unit Status and Echo-back	2 bytes
+66	Configuration ID	2 bytes
+68	Unit Conduction Time	4 bytes
+72	Reserved	20 bytes
+92	Present Error Information	12 bytes
+104	Error Log Count	1 byte
+105	Operation Log Count	1 byte
+106	Error Log	40 bytes
+146	Operation Log	40 bytes

When polled the G9SP will send 146 bytes of data to the PLC. D203 contains the first two bytes of data and D204 is the next two.

This charts shows the order in which data is read and written into PLC memory from the G9SP. For a more detail list of each data point look at the PLC I/O Excel file.

SYSMAC always in control

G9SP RS-232 Communication PLC Code Explained

This explanation is based on demo code in the G9SP Operations Manual Section 7-2-2

G9SP RS-232 Communication PLC Code Explained

The PLC code for RS-232 communication to a G9SP can be broken into four parts:

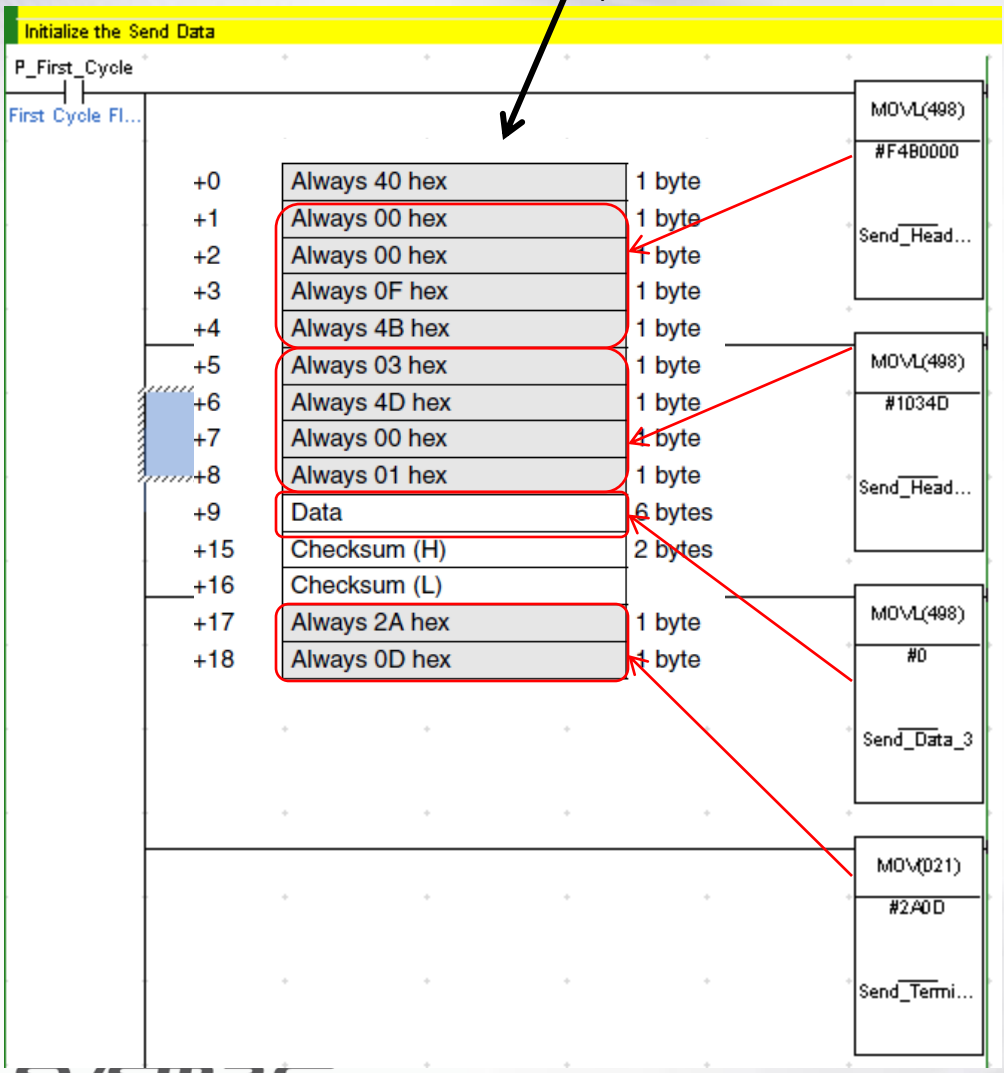
1. PLC first scan initialization
2. RS-2323 Communication send command
3. RS-232 Communication receive command
4. Communication checksum calculations

This document will explain how the PLC works for each of these section of PLC code. This explanation is based on demo code in the G9SP Operations Manual Section 7-2-2

PLC Initialize First Scan Code

Only the very first scan of the PLC will the code do this step, this is only done when the PLC is first powered up or when put back into run mode.

This section of code is ran on the PLC's first scan and sets up the data for the communication Send command words



This function sets up the first four bytes in the communications command.

This function is the next four bytes in the communications command.

This function writes 0's to the G9SP Input Optional Com registers

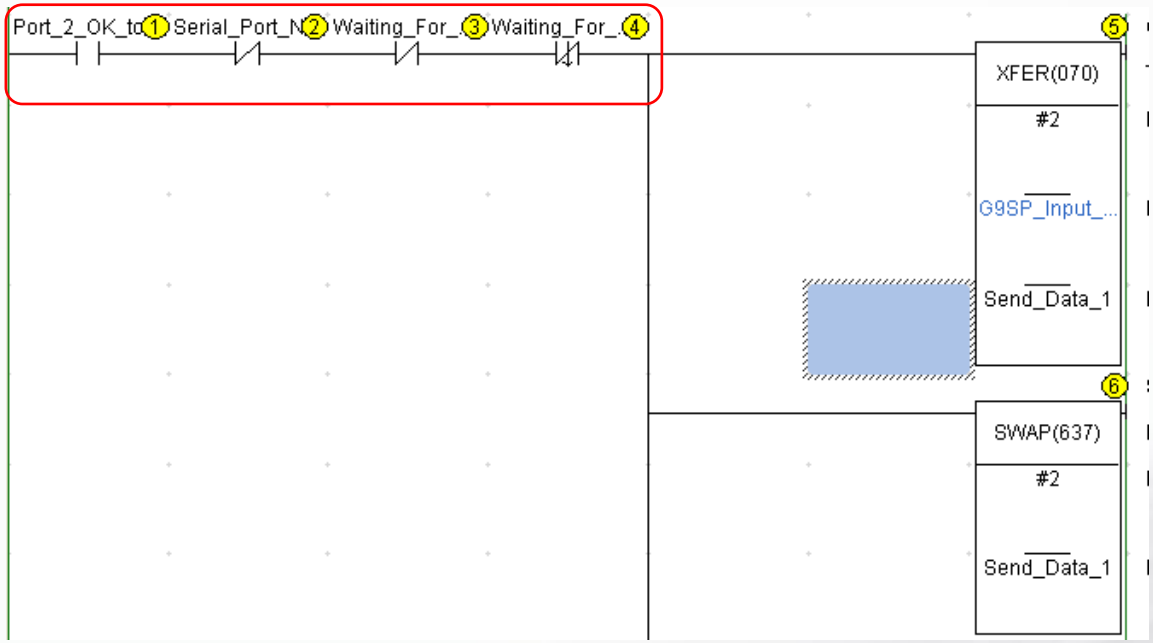
This function sets up the last four bytes in the communications command.

PLC Main Code Send Command


```
Send
1   If the port is available to send data
2   and the port is not resetting
3   and the program is not already waiting for a response
4   and this is not the scan when a response was recieved
5   Copy the data from CIO 200
6   Swap the 4 bytes of data for proper alignment in the serial string
7   Make the loop counter 7, which is 7 words of recieved data across which the checksum needs to be calculated
8   Set the pointer to &100, which then points to D100
9   Call the Checksum Subroutine
10  Move the Calculated Checksum into the position to send out in the serial string
11  Send the string
12  Mark the fact that the system is now waiting for a response
```

The PLC's send command will excite the logic in these steps.

The first four steps of the PLC send code is checking the status of the communications port, if the port is ready to send data the code moves to the next step



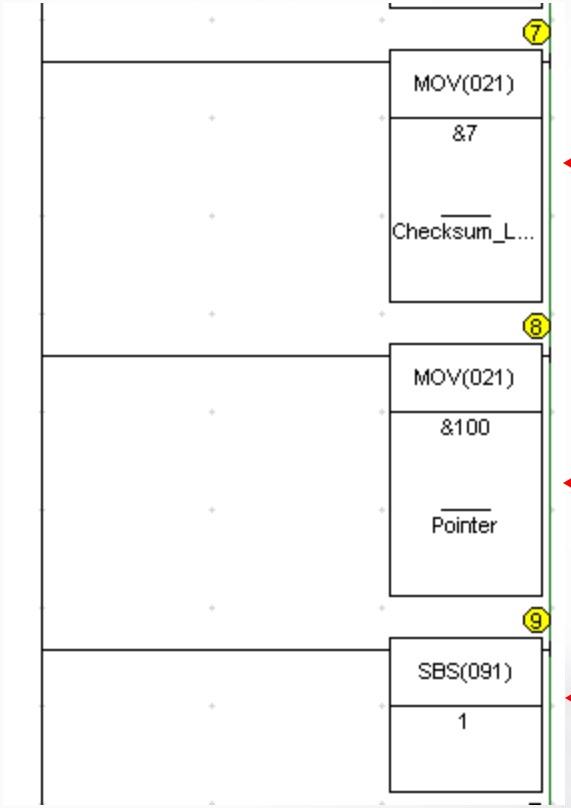
The function block in step five moves the two words of G9SP optional input communications bits from the holding register to the send command register.

PLC Main Code Send Command

Send

1 If the port is available to send data
2 and the port is not resetting
3 and the program is not already waiting for a response
4 and this is not the scan when a response was recieved
5 Copy the data from CIO 200
6 Swap the 4 bytes of data for proper alignment in the serial string
7 Make the loop counter 7, which is 7 words of recieved data across which the checksum needs to be calculated
8 Set the pointer to &100, which then points to D100
9 Call the Checksum Subroutine
10 Move the Calculated Checksum into the position to send out in the serial string
11 Send the string
12 Mark the fact that the system is now waiting for a response

The PLC will excite the logic for these step in this section



This function moves loop count for the checksum subroutine. This is the number scans the subroutine will run, see Checksum section for more details.

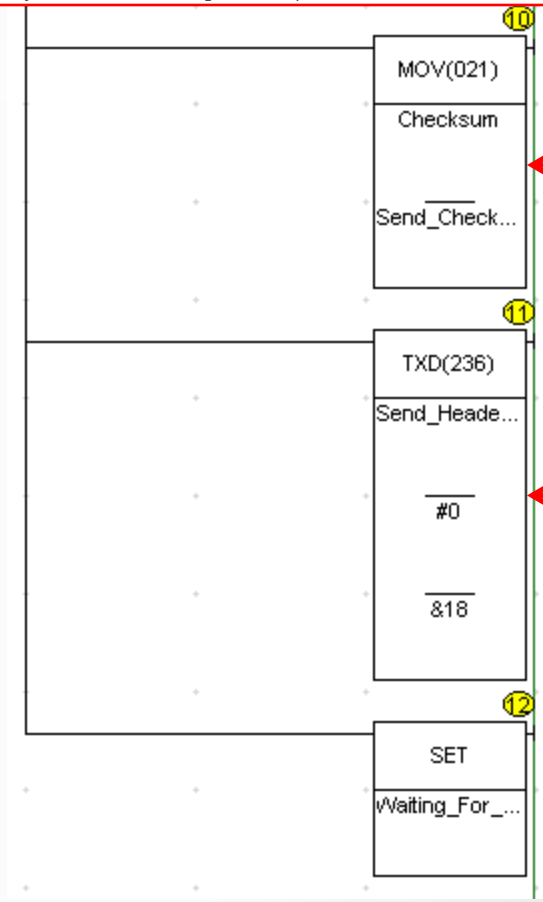
Load a pointer for the checksum logic. For this PLC code example we are using D100 as the first DM register see Checksum section for more details

This function block start the Checksum subroutine

PLC Main Code Send Command

Send

1 If the port is available to send data
2 and the port is not resetting
3 and the program is not already waiting for a response
4 and this is not the scan when a response was recieved
5 Copy the data from CIO 200
6 Swap the 4 bytes of data for proper alignment in the serial string
7 Make the loop counter 7, which is 7 words of recieved data across which the checksum needs to be calculated
8 Set the pointer to &100, which then points to D100
9 Call the Checksum Subroutine
10 Move the Calculated Checksum into the position to send out in the serial string
11 Send the string
12 Mark the fact that the system is now waiting for a response



The PLC will excite the logic for these step in this section

This function moves the calculated checksum value from the holding register to the send command register to be sent out with the send command.

This function moves the calculated checksum value from the holding register to the send command register to be sent out with the send command.

SYSMAC always in control

PLC Main Code Receive Command

Data Reception

1. If 198 bytes were recieved, which is correct
2. Recieve the data from the port
3. Make the loop counter 97, which is 97 words of recieved data across which the checksum needs to be calculated
4. Set the pointer to &200, which then points to D200
5. Call the Checksum calculation subroutine
6. If the calculated checksum matches the recieved checksum
7. and the terminator was in the correct position
8. Swap 5 bytes of data to align them correctly
9. Extract the 32 bits of optional comms data and move them to CIO 202
10. Increment a counter for OK Comms Cycles
11. If the calculated checksum does not match the recieved checksum
12. or the terminator was not in the correct position
13. Increment a counter for NG Comms Cycles
14. Reset the Waiting for Response bit

The PLC will excite the logic for these step in this section

This first step for the receive command count the number of bytes the communication port has received, this function only turns on when the PLC has received 198bytes of data.

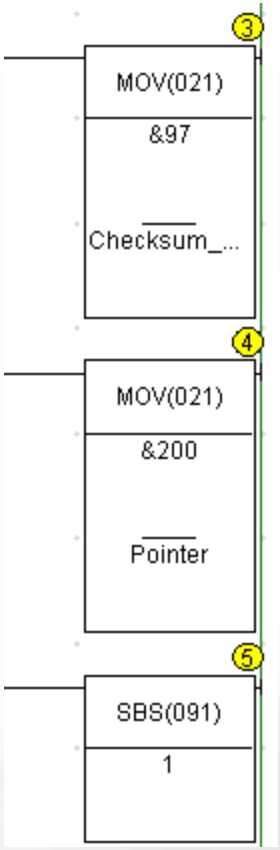
This is the PLC receive function block, this block moves 198 bytes of data to the PLC DM registers starting with DM200.

PLC Main Code Receive Command

Data Reception

1. If 198 bytes were recieved, which is correct
2. Recieve the data from the port
3. Make the loop counter 97, which is 97 words of recieved data across which the checksum needs to be calculated
4. Set the pointer to &200, which then points to D200
5. Call the Checksum calculation subroutine
6. If the calculated checksum matches the recieved checksum
7. and the terminator was in the correct position
8. Swap 5 bytes of data to align them correctly
9. Extract the 32 bits of optional comms data and move them to CIO 202
10. Increment a counter for OK Comms Cycles
11. If the calculated checksum does not match the recieved checksum
12. or the terminator was not in the correct position
13. Increment a counter for NG Comms Cycles
14. Reset the Waiting for Response bit

The PLC will excite the logic for these step in this section



This function block is part of the checksum subroutine, this is the loop count for the checksum calculations. See Checksum section for more details

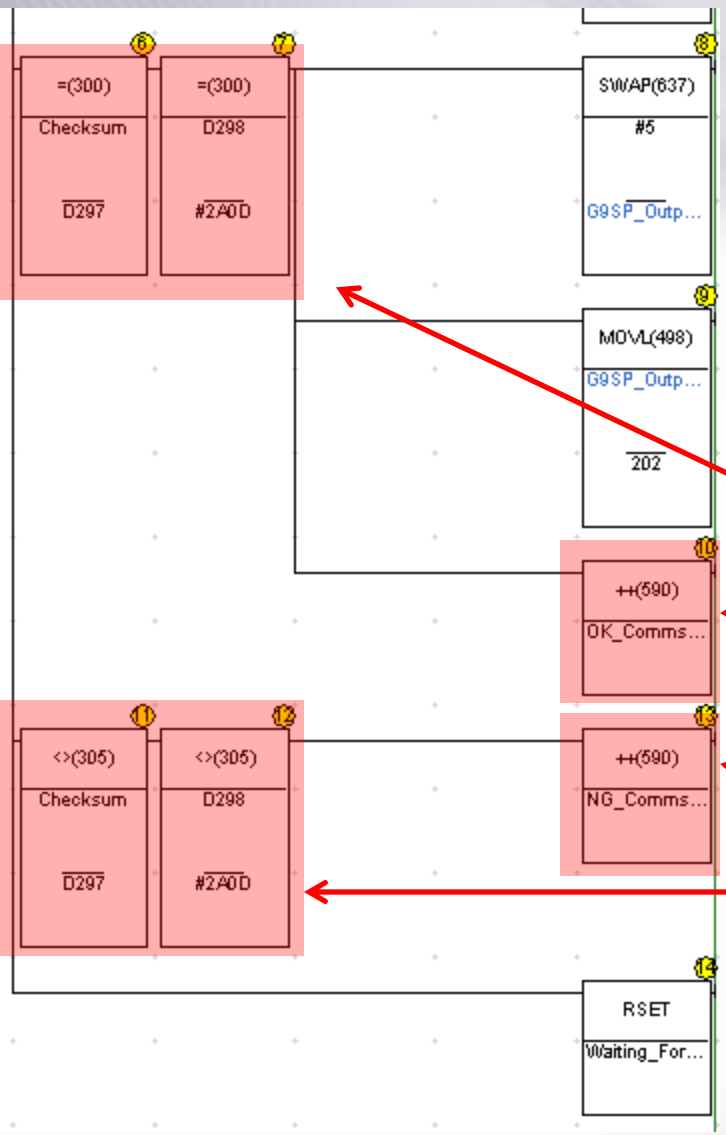
Load a pointer for the checksum logic. For this PLC code example we are using D200 as the first DM register, See Checksum section for more details

Run the Checksum subroutine, See Checksum section for more details

always in control

PLC Main Code Receive Command

OMRON



Data Reception

1. If 198 bytes were recieved, which is correct
2. Recieve the data from the port
3. Make the loop counter 97, which is 97 words of recieved data across which the checksum needs to be calculated
4. Set the pointer to &200, which then points to D200
5. Call the Checksum calculation subroutine
6. If the calculated checksum matches the recieved checksum
7. and the terminator was in the correct position
8. Swap 5 bytes of data to align them correctly
9. Extract the 32 bits of optional comms data and move them to CIO 202
10. Increment a counter for OK Comms Cycles
11. If the calculated checksum does not match the recieved checksum
12. or the terminator was not in the correct position
13. Increment a counter for NG Comms Cycles
14. Reset the Waiting for Response bit

If the calculated checksum is correct the PLC will send a coms ok command

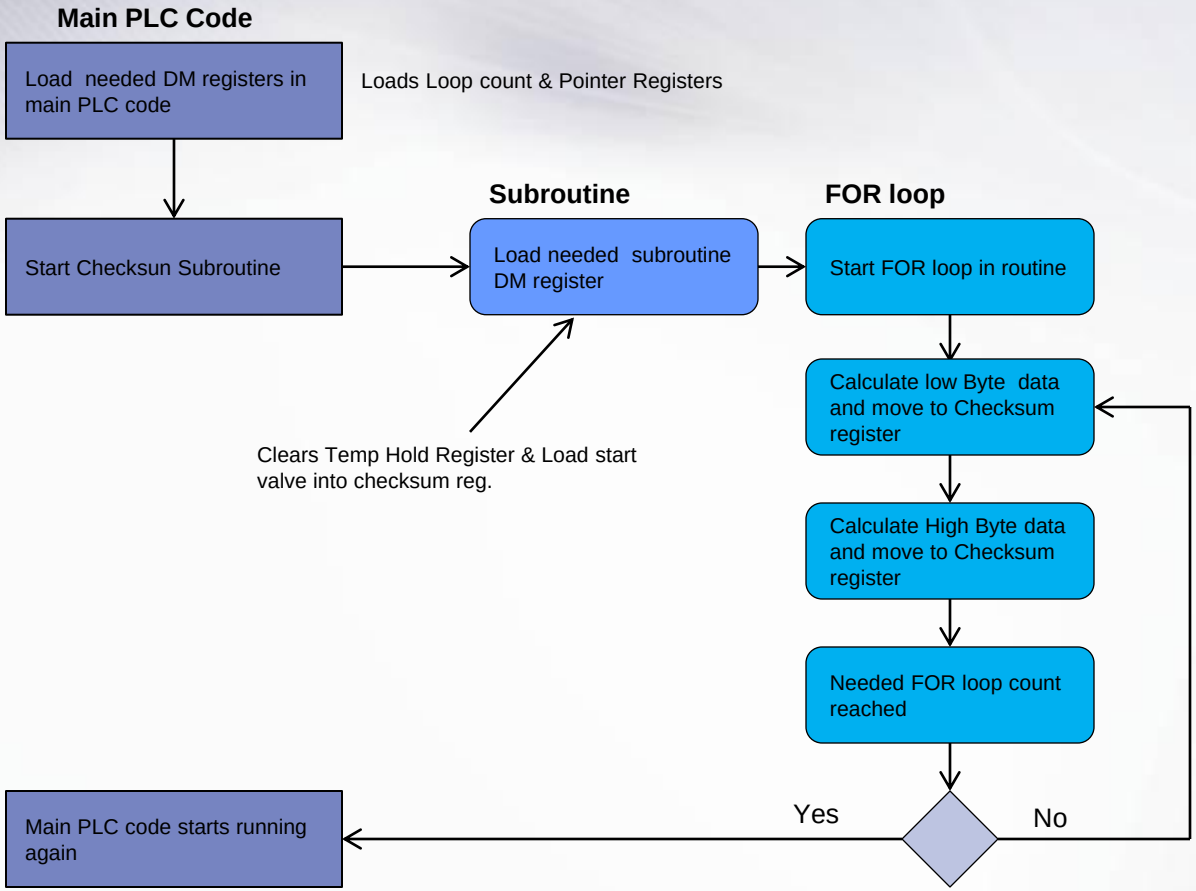
If the calculated checksum is not correct the PLC will send a coms NG command

SYSMAC
always in control

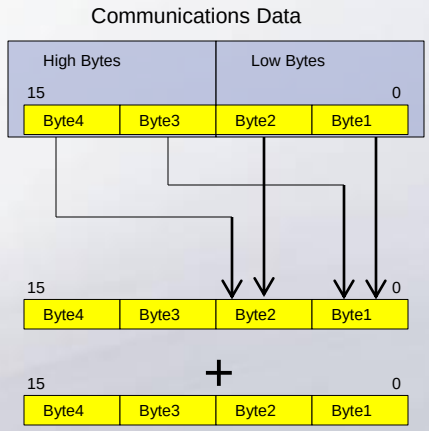
G9SP RS-232 Communication CheckSum Explained

This explanation is based on demo code in the G9SP Operations Manual Section 7-2-2

CheckSum Operational Flow



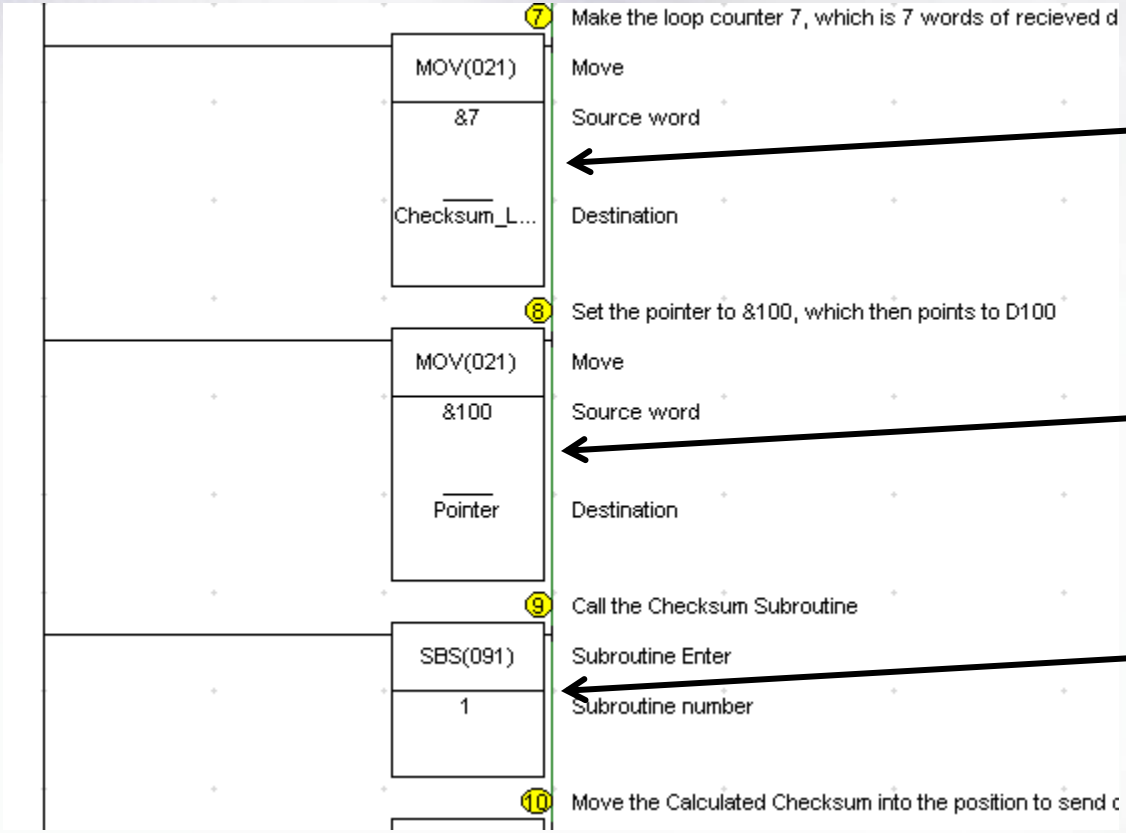
This code will use a number of registers to do a checksum calculations, for this example these registers were setup:
Checksum DM6730
Checksum Loop count DM6731
Checksum Pointer DM6732
Checksum Temp Hold Register DM6733



The checksum is calculated using this method, the low bytes form the communication data is move to a temporary holding register and then added to the checksum . The high bytes are moved to the temporary holding register and then added to the checksum. Once this is done the code will increment the com data to the next data point and repeat this step. The FOR loop count will determine how many times it repeats this step.

G9SP CheckSum Code in Main PLC Program

This is the CheckSum for a Send command, receive is a little different from the send



The first step is to load 7 into the Checksum loop count. If this was the receive command you would load 97

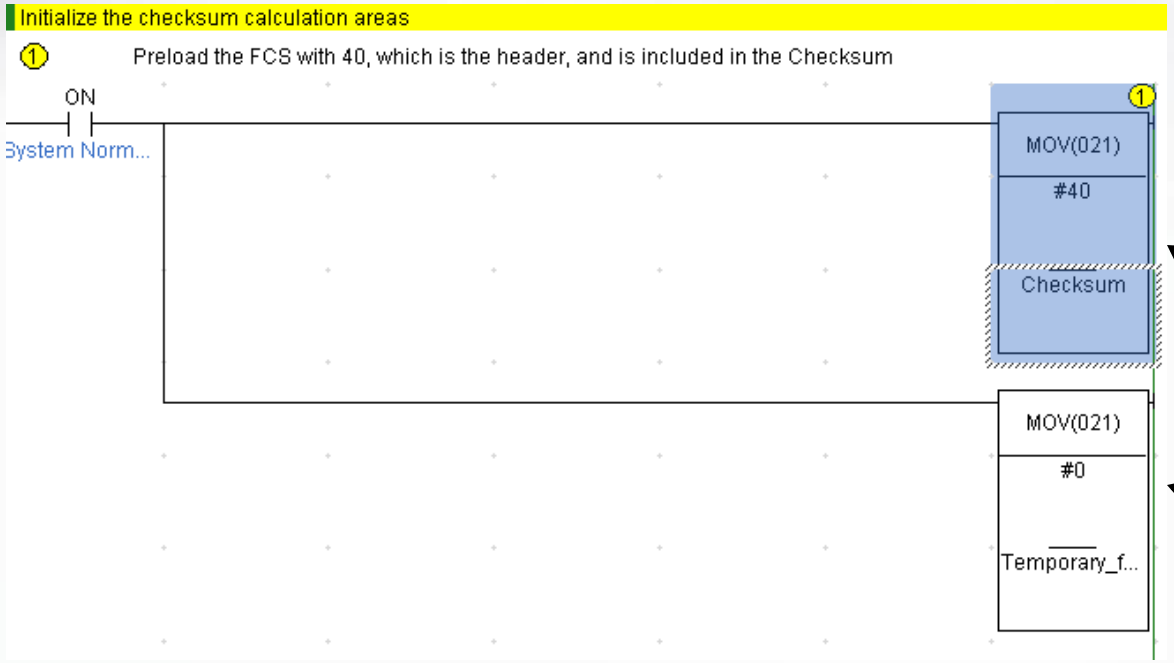
Load a pointer for the checksum logic. For this PLC code example we are using D100 as the first DM register

Once these two number are loaded run the Checksum subroutine

All these function are done within the main PLC code.

G9SP CheckSum Code in Subroutine

This is the CheckSum code running in the subroutine, once the subroutine has stared the PLC will not run any part of the main communication PLC code section until this routine is done.

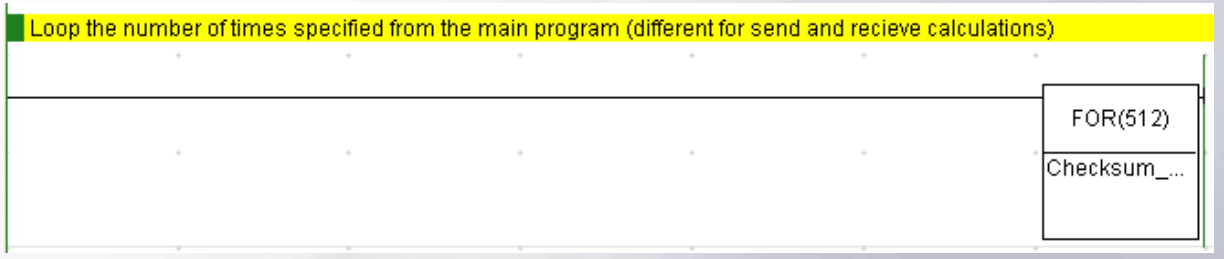


Once in the Subroutine two values need to be Initialize, Checksum Valve and the Temporary holding register for checksum.

Set the starting value in the checksum as #40 Hex. This is done because 40 is part of the header and is used as part of the checksum calculations.

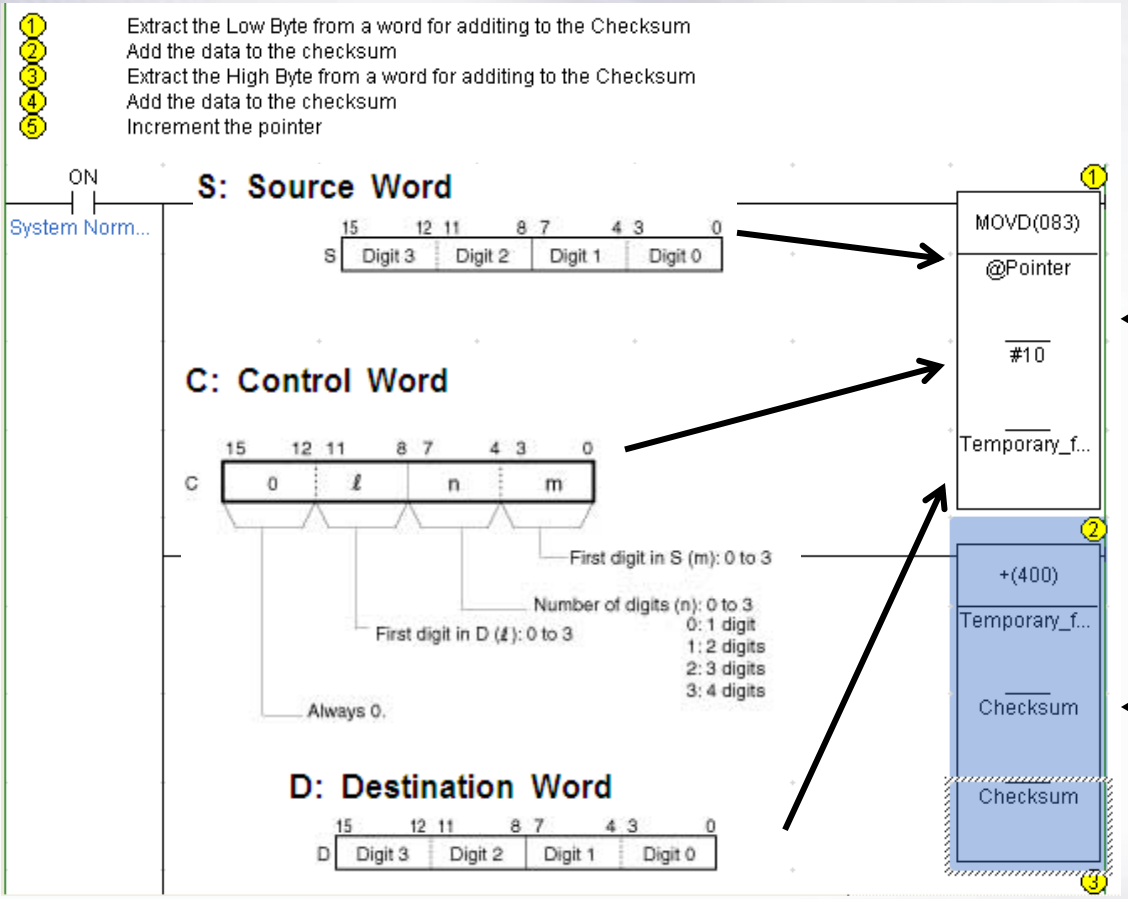
The Temporary Checksum register is loaded with 0. This value will be incremented within this routine as it runs

The next step is to set up a FOR loop in the subroutine, this part of the code will take the number that was loaded into the Checksum loop count register (7 for send command 97 for receive command) and only run this part of the code for 7 PLC scans or 97.



G9SP CheckSum Code in Subroutine

This part of the code will run in the FOR loop and the PLC will only use this part of the code for the set number of PLC scans that was loaded into the Checksum Loop count memory register.

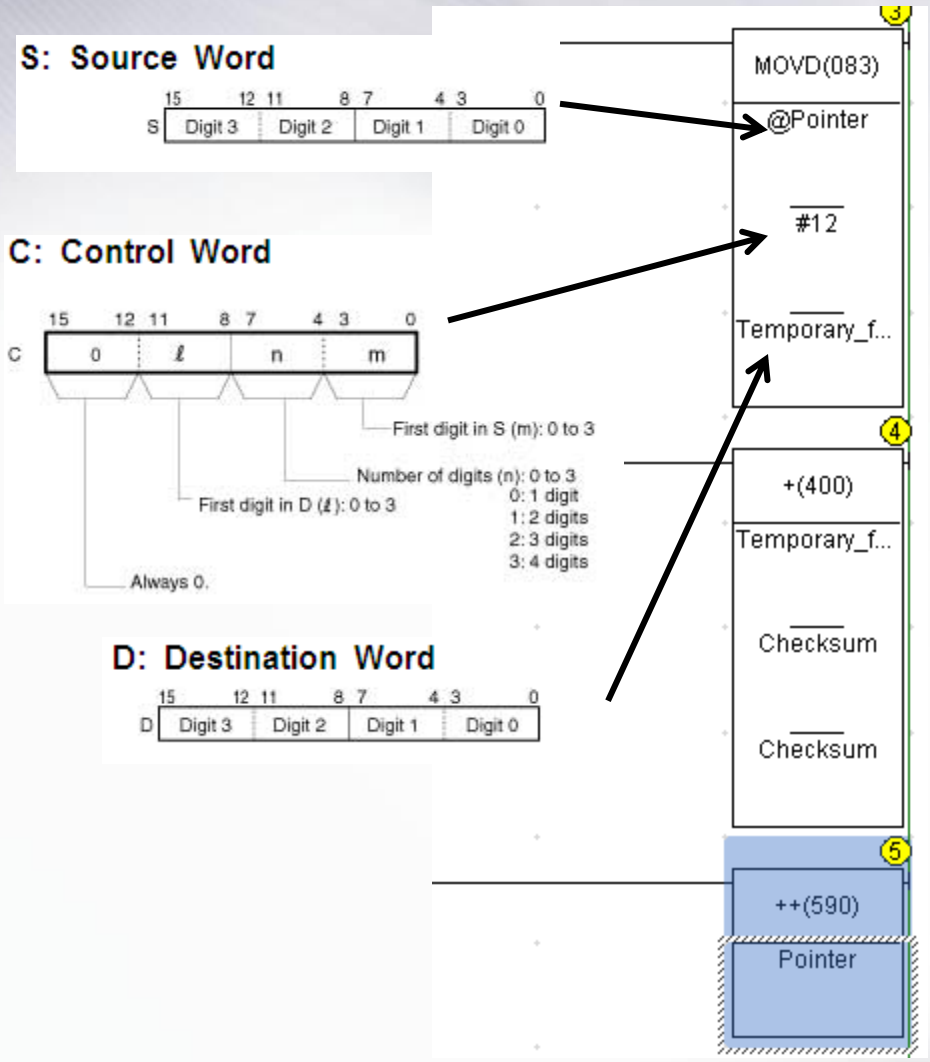


The first step in the FOR loop is to extract the Low Byte data of the first word and add this to the checksum

This function put the data into the correct location of the holding register

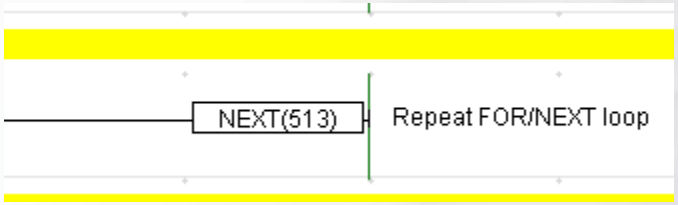
This function takes the number in the holding register and adds it to the checksum total

G9SP CheckSum Code in Subroutine



This step is just like the last step but this time we are moving the high Byte data and then add it to the checksum total

In this step the pointer is also incremented by 1. This will move the pointer to the next communications data point.

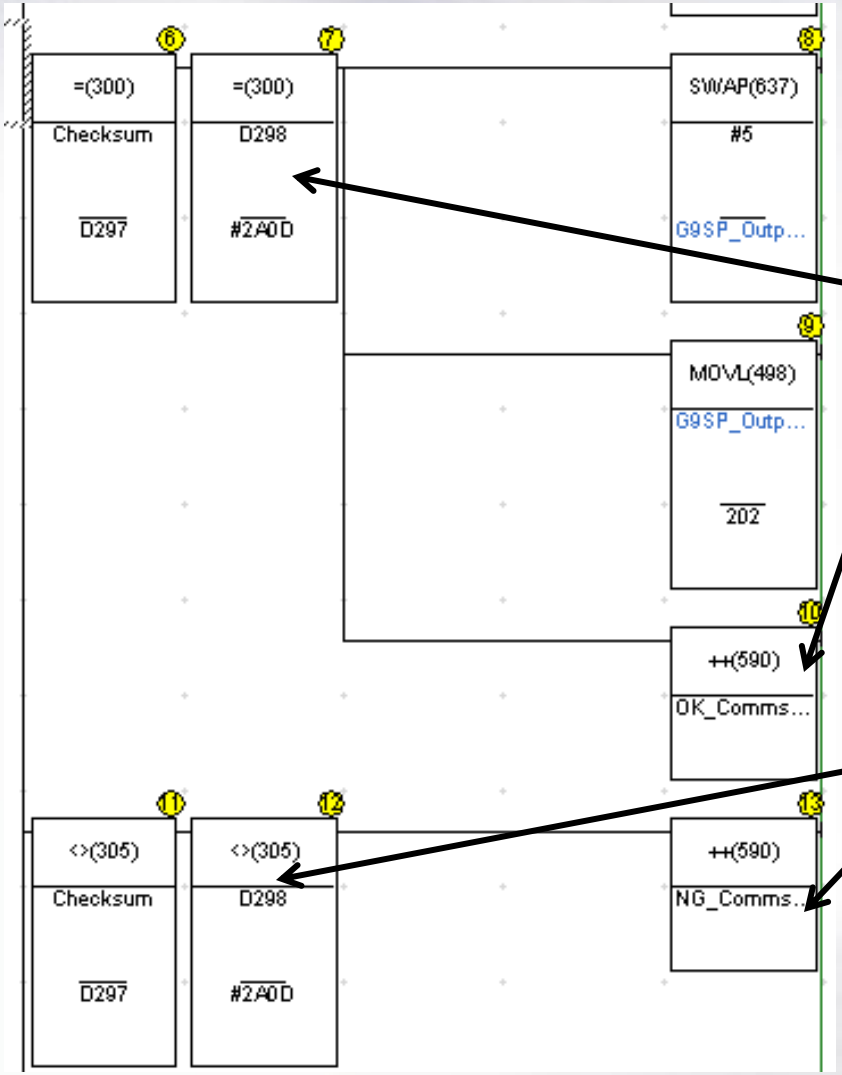


The code will at this point check to see if the FOR loop has reached the needed total number of scans, if has not reached the needed number it will repeat checksum calculations until it does

SYSMAC always in control

G9SP CheckSum Code in Main PLC Program

Once the code has reached the needed number of loops it will jump back to the main PLC code.



Once back in the main PLC code the PLC will check it see if the calculated checksum is correct.

If the calculated checksum is correct the PLC will send a coms ok command

If the calculated checksum is not correct the PLC will send a coms NG command

SYSMAC always in control