SNP PLC Program

SNP Program Architecture

All machine code is separated into folders per machine, from there each machine has one Main Database labeled M#\_DB along with all non-connection/ line based functions such as the message sender IO Processor and a folder containing the Packet Functions and data blocks. Each packet function has a matching data block with the packet setup information.

Since all machines talk back through one connection while each Message sender generates their own packets they talk to the line functions to send the packet. Other than that, tie no logic goes from within a machine structure to outside of the machine structure.

All IO is setup so that from 10.0 until 19.7 all inputs and outputs are assigned to Machine 1 with each consecutive machine taking the next range of 10 inputs. Below is the inputs and outputs for machine1. These inputs and outputs are copied over the rest of the machines.

Inputs

M1RunningStatus %I10.0 Running Status High when on Low when off

M1OutputGood %I10.1 Output Good Set high on good output

M1OutputBad %I10.2 OutputBad Set high on bad output

M1EmptyHead %I10.3 Head Empty set high when current head is empty

M1Index %I10.4 Index set high on an index

M1StopButton %I10.5 Set high when the stop button is pressed

M1AlarmStop %I10.6 set high when the alarm goes off

M1HeadNumber1 %I10.7 Set high when Head 1 is being used

M1OtherError %I11.0 set high when it is a miscelanous part error

M1ErrorPin0 %I11.1 error pin0

M1ErrorPin1 %I11.2 error pin1

M1ErrorPin2 %I11.3 errorpin2

M1ErrorPin3 %I11.4 errorpin3

M1ErrorPin4 %I11.5 errorpin4

M1ErrorPin5 %I11.6 errorpin5

M1ErrorPin6 %I11.7 errorpin6

Outputs

M1StackLightG %Q10.1 Machine Power Enable

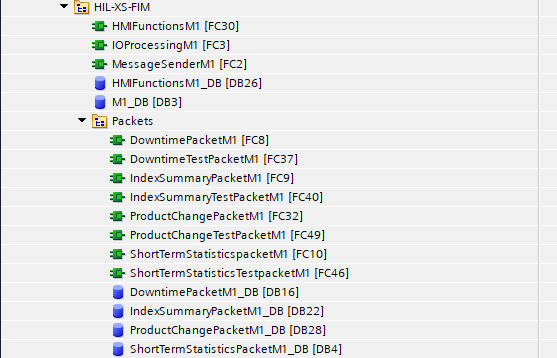
M1StackLightB %Q10.0 SNP Machine in running state

M1StackLightO %Q10.2 Index Pulse

M1StackLightR %Q10.3 Unscheduled Downtime

Program Blocks

For all intents and purposes M2-4 are just copies of M1. As such I am going to explain M1’s setup and it can be assumed the rest follow suit.

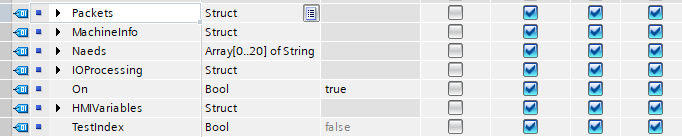


For XS the First Machine is the FIM. The top block is the HMI Functions Block. This block contains all logic relating to the Two HMI’s. for the most part it is the same logic driving both HMI’s with the exception of the logon logic as otherwise it would prompts both screens to go into administrative mode when one logged on.

Next is IO Processing. This contains All Output assignment as well as all logic handling error gathering, Index logic, and machine tracking.

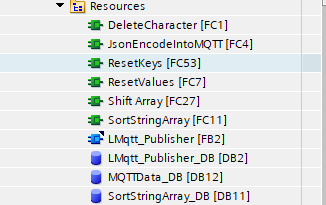
Next is the Message Sender block. This block is a collection of rungs dedicated to each packet type the machine supports. On the X packet send bool being set high it will send X packet through MQTT to the ActiveMQ broker setup in the Resources section. This handles waiting for the block to become free and ensuring delivery however it is limited in that if it receives two pulses before the first resolves it will not register the second in order to prevent overwriting the data being sent.

Next is the HMI Functions Data block that contains temporary variables to do with the HMI Functions block.

Next there is the M1\_DB data block. This is the Main data block for each machine and contains all data relating to the machine, whether it be the packet types what their statuses are how many good parts were produced etc. 

All data relating to packet sending and status is in the Packets Struct while all data on the last packet and current Machine Info ( like good part bad part counts or any errors set) are set in Machien Info. Next Naeds contains a the list of naeds IO processing contains Onshot variables for the IO Processing function and HMI Variables contains variables that are accessed by the HMI’s.

Next there is a collection of Packet functions and datablocks. Each datablock contains the pairs used as well as how many of those paris are used and the header bytes for the packet. The function block contains the logic for seting up the JSON pairs before they are sent to the packager.



Next is the Resource group.

This group is a collection of assorted resources used by all machines and the line functions. These are mainly small SCL functions designed for a set purpose.

Delete Character will go through a string passed in and remove any instance of a character from it ( used to remove spaces from ints that were converted to strings.

JsonEncodeIntoMQTT encodes a pair array into JSON then inputs it intothe MQTT message buffer.

Reset keys sets the Key value of a pair array betwean the start and end value ( used by the apcket generators)

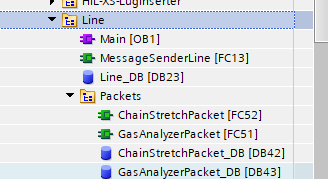
Reset values does the same but with the value section.

Shift Array Shifts an array up or down by one and wraps the top or bottomof the array respectivly.

LMQTTpublisher is the function that handles talking to MQTT. Along with its matching datablock.

MQTTData\_DB is the data blockcontaining all user set information such as the username password topic and broker ip.

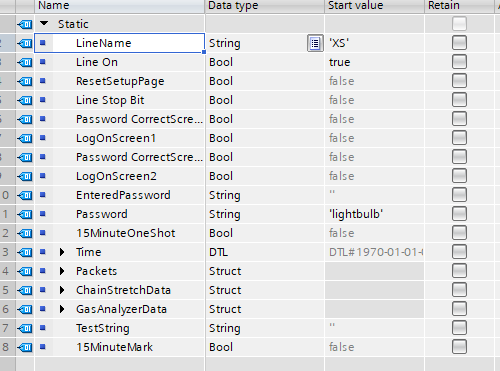
Sort string array sorts an input array alphabetically.



The Line group contains all Line based logic including the entry point for the program. Inside of main it handles Logon 15 minute index sending, as well as enabling each machines IO processing, Message sending and HMI Functions.

Message Sender line is a copy of the machine message sender but configured to send ChainStretch and Gas Analyzer packets which should be a by Line packet.

Line\_DB contains all line based data for the SNP machines.



This includes the name whether the setup is on weather the password was correct when entered what the entered password is what the correct password is the 15minute oneshot variable current plc time and the 2 packets handled by the line and the data surrounding them.