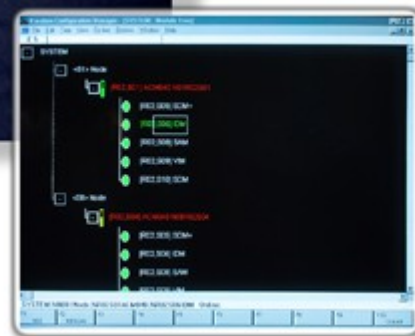
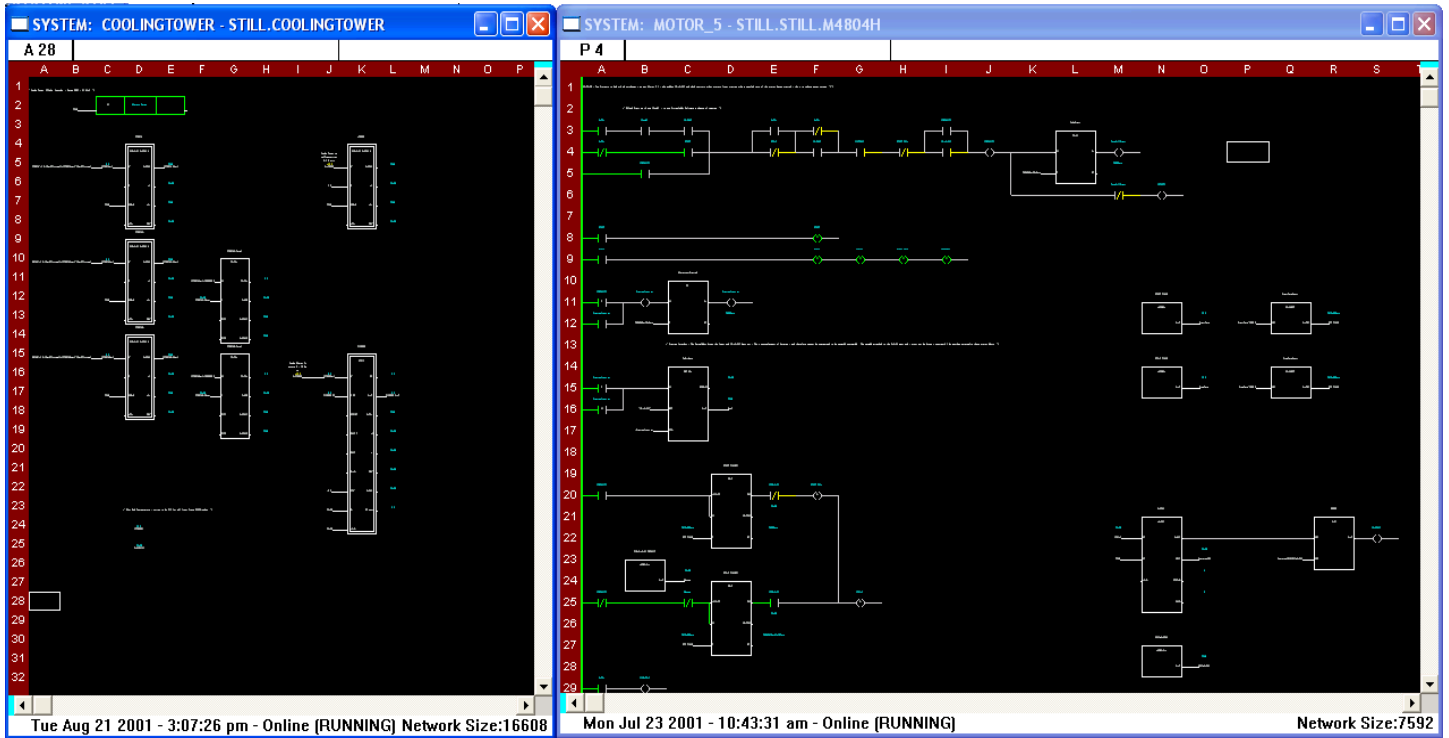


APACS 4mation



Military grade housing – you could drive a tank over them and they would survive
– the outside casing of each module is a heatsink for the components inside - so you you'd never have a fan blowing dust borne air directly over PCB components.

4mation comprises 4 languages

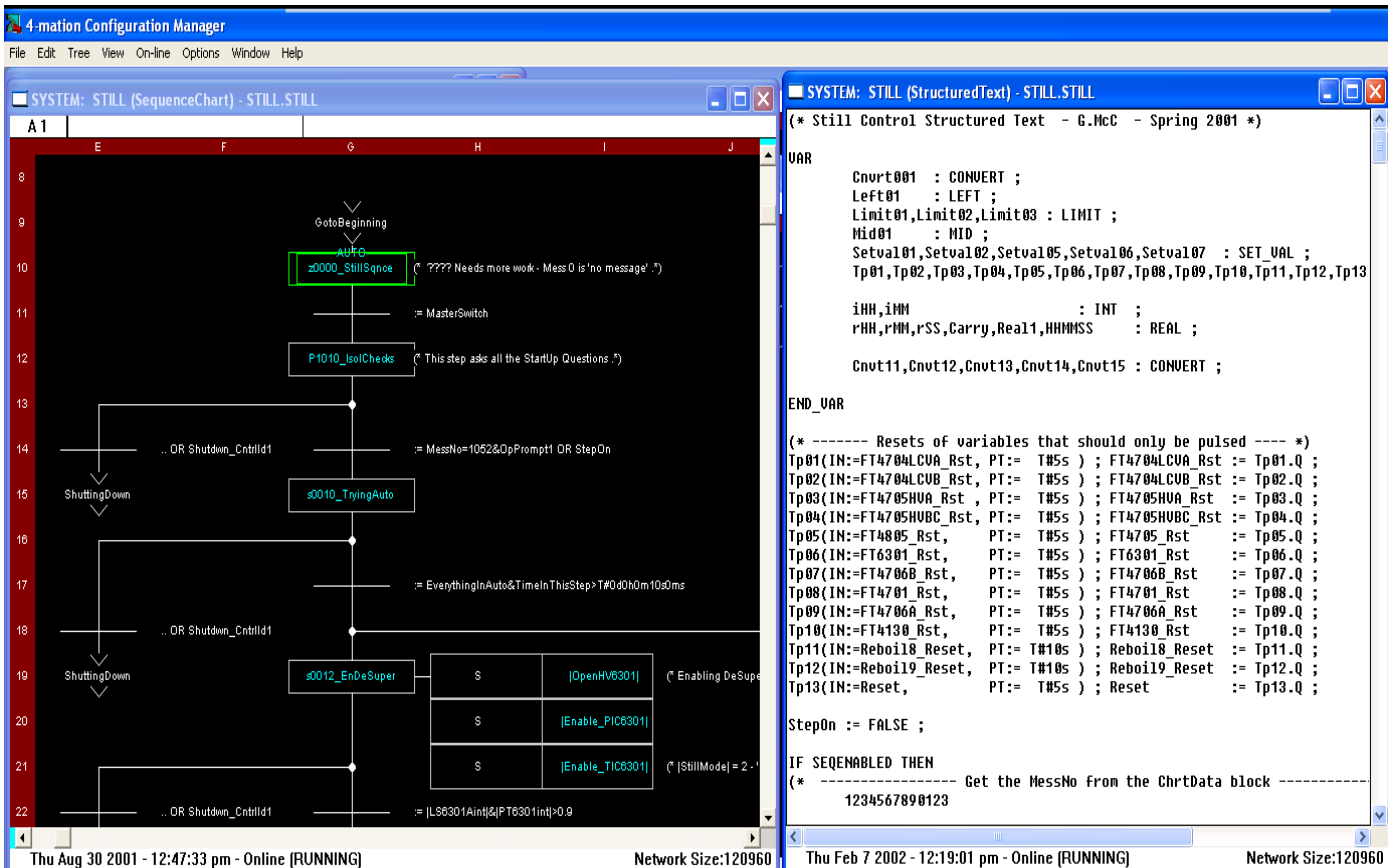


Function Blocks

With an 'Action' Block borrowed from the Sequence Chart language.
The Action blocks allow sheets made in different languages to occupy the same space and share the same variables.

Ladder Logic

with 'Function Blocks' borrowed

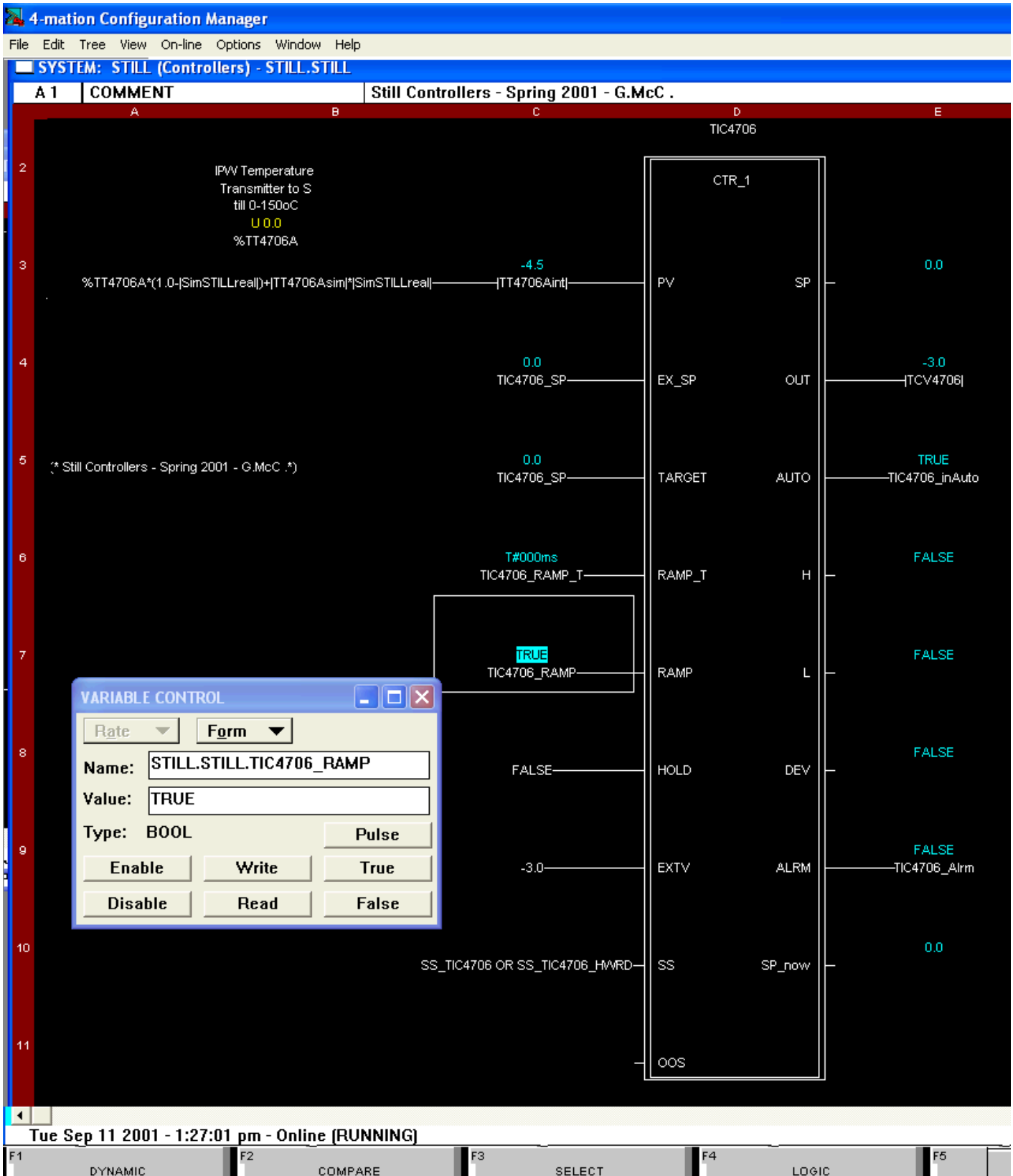


Sequence Chart

With 'Action' Blocks and Boolean decisions

Structured Text

Where providing you declare them you can put Timers, Convert blocks etc ...



This graphic shows :-

- CTR_1 (Controller (Type 1)) with external and internal ramp capability.
- Real World variable %XXXXX (with Yellow (BAD) current value)
- Global Variables such as |TT4706int|
- Variable Control pop-up showing control of TIC4706_RAMP (Cyan background = Variable colouring inversed) - Easy Forcing of variables – doesn't affect the Time Stamp of the sheet.
- Structured text on a Function Block sheet "SS_TIC406 OR SS_TIC4706_HARD"
- Shows how a time variable is defined T#47d23h59m59s999ms
- Comments can be placed anywhere on a screen surrounded by (* Comment *)
- If you want to add an extra nub (for example) you need to delete every use of the Function Block on the sheets (or create a new Function Block) - the existing code attached to the nubs remains in tact – it remains on the sheet, waiting for the new version of the function block to be dropped back down.
- The system protects you from bad coding practice it won't attach an Integer in one cell (A1) and to a Real in cell B1, where as if they were both the same type of variable then they would be connected.
- The graphic also shows that not every nub needs to have a variable attached – Input or Output.

Choose the language that is most suited to the requirements,
Use Ladder for discrete control
Use Sequence Chart for batch and sequence processing
Use Structured text for low level control such as Operator Prompts and
Calculations, and compressing otherwise chunky code such as the operation of the Klaxon
And use Function block for everything else.

The languages are not set in stone and you can use function blocks on a Ladder page
And Structured Text (on a single line) on a Function Block Page OR Ladder sheet.
The Actions from a sequence chart can be incorporated into the
Function Block chart – to extend the size of each sheet.

The time stamp in the bottom left hand corner tells you when sheets are updated –
and the Network size in the right hand corner shows you the size of the network – which ultimately has
An impact on the scan time of the ACM (processor).

Today I am demonstrating the APACS 4mation software to show you how good it is and show you how the architecture of the software matches the robustness of the physical modules. Talk about “Fit for Purpose”.

The people who developed this software really knew what they were doing when they designed the software – they obviously had experience of industrial software in the field. It wasn't written by a load of geeky students who just thought it would be a good idea. This is excellent software right from the first moment you see it. And I just hope that Siemens have incorporated at least some of the APACS structure into their latest offering. Otherwise - within the next 10 minutes you'll see what the industry has lost when Siemens bought Moore Products to close out the competition back at the turn of the Millennium.

APACS 4mation ...

- There are two versions of the main Controller - an ACM (Advanced Controller Module) and a QUADLOG Controller – High Integrity controller.
- Military guage housing on all modules (external casing of each module having fins on it to incorporate heat sink capabilities – very high dust tolerance – as you should never have a fan blowing cooling air directly over PCB components.)
- Hardware page can be drilled down straight into assigning IO.

4mation advantages

- Dropping a function block onto a sheet automatically assigns all the variables that block will need (including internal timers) – the programmer doesn't need to worry about variables and their allocation – the software takes care of them.

VARIABLES

- BOOLEAN, INTEGER, REAL , STRING and TIME all have their own attributes that are accessible all the time. If you want to see the quality attribute of a variable – you can specify the variable as the input to a standard “QUAL” block which will provide you with the quality of the input variable – but that is actually a long handed way of getting the quality of a variable – you can just address the quality attribute directly – “LIT4706.PV.q” And the “quality” attribute moves through the system so if a Bad IO (AOR for example) is attached to the PV nub of a function block then the PV nub becomes BAD, and any maths done using that bad input will also assign the bad quality to the resultant variable. (not in Structured Text).
- Variable forcing using a mini popup.
- you can force variables inside a function block that is used multiple times – knowing that you are only forcing this particular instance's variables.
- Enable!Disable buttons allow the variable to be locked from changes from within 4mation including the real world and any comms such as a SCADA. This means that forcing doesn't affect the time stamp of the sheet.
- You can force any variable – A Real word variable whether it's an input or an output (%) , a global variable (!global!) , you can force any variable within a Function block – it doesn't matter what type it is.
- Visibility – using Ctrl I (in) and Ctrl O (out) - zooming to any level is possible on any sheet (apart from the Structured Text sheet). Gives a very easy way of getting an overview of the software.
- One sheet can be extended using sheet extenders (which are actually Sequence Chart 'Actions') - so that one sheet can have many sheets of different languages controlling a particular aspect of the plant. All the variables on those sheets are local to each other. The control of a whole of a process area (such as a fermenter \ filter etc...) can be contained on one sheet.

- You can have multiple Simulators running on the same VM so that you can check inter-reactions.
- Variables are part of the structure of the language – they can be accessed (from SCADA and elsewhere) by their location ACM_AAAA,FilterXX,TIC4706,RAMP_ENABLE is the Boolean variable controlling the RAMP functionality of the temperature controller loop for FilterXX in ACM_AAAA
- Standard Function blocks are represented by a single border line (unless they are extendible – in which case their bottom border line is a double line whilst the remainder of the block is drawn with a single line).
- User Defined Function Blocks have a Double border line all the way around them – so they are easy to distinguish from the Standard Function Blocks.
- The Standard Function Blocks can be extended up to 16 nubs on their inputs.
 - 'AND' blocks, 'OR' blocks, CONVERT blocks and similar functions can have multiple inputs,
 - a MOVE block can move multiple things on the same trigger. It can move a couple of Reals to other Reals and then an Integer to an Integer, and being controlled by the same block – their movement happens at the same time.
 - There is also a SELECT block – which is a logical switch – “either move this thing or this thing to the output depending on the value of the [EN] (enable) nub.” The things can be any type of variable :- BOOLEAN, REAL, INTEGER, STRING & TIME (not arrays though)
 - Setpoints and control parameters can be SETVAL into arrays and stored in an excel spreadsheet – to retain their setting between downloads of configurations – this was developed by me (Greg McCormack) and ended up as a very useful tool being used by Siemens-Moore (in Bolton (at the time)).
- In the Ladder logic – proper ladder logic can be incorporated with proper latching capabilities, since variables are correctly allocated and handled by the operating system.

SCADA IMPLICATIONS

Easy variable addressing within the PLC has implications for other systems as well – the SCADA ends up being structured – the setpoint for a controller – has an obvious address ...

The address for TIC4706_PV is ACM_AAAA.FILTER001.TIC4076.PV

You can even address the quality bit of the variable from SCADA just as easily ...

ACM_AAAA.FILTER001.TIC4076.PV.q

And (as mentioned elsewhere in this introduction) the Quality attribute is moved with the variable

– so if (and when) the real world PV variable is BAD – and is attached to the [PV] nub of the controller TIC4706 – then the nub's quality attribute also becomes BAD.

CONCLUSION

The choice of being able to choose the best language to suit the requirements is a huge benefit. Ladder logic with PID controllers – don't need to have their own OB35 – as the system is fast enough and automatically will adjust itself to make sure that scan time changes will not affect the performance of the fastest PID controllers. If you need faster then you should be doing the PID control in a dictated bit of kit.

The removal of the responsibility for many of the aspects of PLC configuration allow you the programmer to concentrate on higher order aspects of the control.

For example – and I am not just blowing my own trumpet – I imagine every PLC programmer in Cougar could do this, within a four year window the whole (well the vast majority 98%) of control system of a complicated factory in Knowsley Industrial Estate, North Liverpool is being controlled by the software of one individual engineer. I was the sole engineer of the site for the vast majority of the time – I was developing the 4mation software and the FIX32 software. The American owners came over to see how well the plant was running – to be able to transfer it over to other sites in the US.

Including a distillation column automatically starting up and shutting down smoothly, with only confirmation operator prompts at critical stages.

Seed Mash with CIP (Cleaning In Place), Six Seed Fermenters, Six Main Fermenters with Oxygen Enrichment, Broth Tank storage with batch information pertaining to the layers, Clarification process, and Cake extraction process using pressure and temperature with the addition of alcohol to form the cake. The cake is then dried in rotating driers and then becomes a powder which then is bagged into different grades – so batch numbering becomes important.

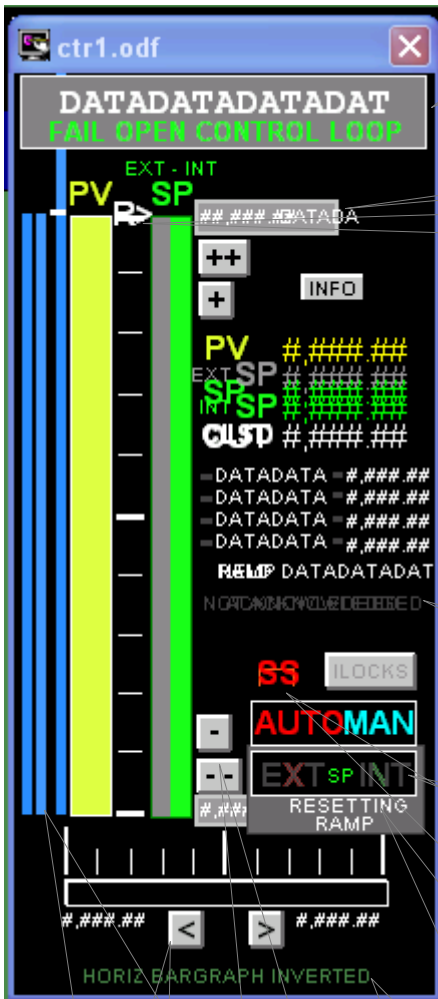
The ancilliary equipment includes chillers, cooling towers, steam house, waste control and drainage, alcohol tank storage (before being refined in the distillation column) – using a sophisticated D,A1,A2,S arrangement with biased weightings, a fermentation process including sterilising with steam at 130oC for 3.5 hours, oxygen enrichment – with all the extra safety considerations that have to be incorporated into the control philosophy when handling dangerous gases. A drying and milling plant – to generate the final product in a dry form.

The broth tanks could store the contents of different batches – so layers were introduced to try and retain the batch details to the final powdered product – it wasn't an exact batch allocation because there is some mixing of the layers – but there are baffles within the broth tanks to try and minimise the mixing of the layers.

There were other occasions where approximations were made as to the volume in tanks – which had a self-learning technique so that it would keep improving on its ability to track the volume in the tank – because it could reset and tweak its estimations itself when it was empty.

From my advantage point now – it is hard to see how all this could have been accomplished in a four year window with just 8 ACMs (PLCs) and without the sophistication of APACS 4MATION.

4mation makes the Top End look sophisticated as well



This single controller faceplate will be suitable for almost all Analogue Controllers – firstly DATADATADATADATA is the name of the controller pulled out of 4mation. (nothing new there)

The ENG-HI and UNITS are specified here – again pulled from 4mation. (nothing new here either)

If the controller is ramping – then the letter R appears to show the Target SP of the Ramp

When in INTERNAL SP – the normal SP disappears and is replaced by External SP and Internal SP – so that you can always achieve a bumpless transfer back into Auto.

There are four standard alarm locations to suit individual requirements of the controller, they could be :-
High High , High and Abs Dev OR
Abs Dev , Lo Dev , Hi Dev and Lo Lo
The alarm types are set in 4mation.

When in InternalSP you can set the parameters of the ramp and then click go and (if necessary) ramp to the Current value external setpoint.

Selecting INT (Internal) splits the SP column into two separate columns and greys out the left hand side to show that the external setpoint is not being used.

SS is STANDBY SYNC – Basically the Interlocks you can click on the ILOCKS button to bring up a TEXT file listing the interlocks for the specific controller

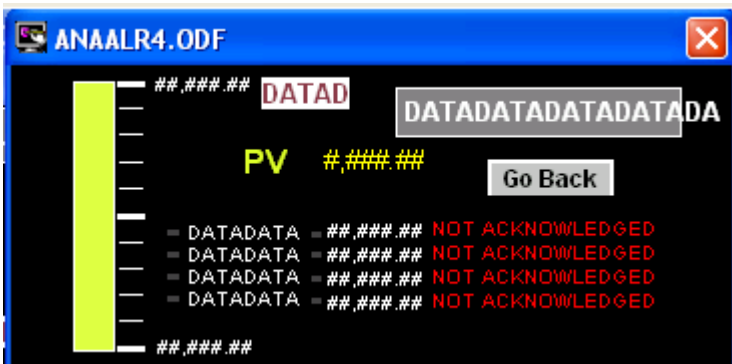
If you need to reset a ramp – then (providing you are in Manual) you can press this button.

For controllers whose output is reversed – such as pressure relief control systems the output bar graph is inverted

When in Manual you can type in an OUT value OR use these buttons.

You can jump the SP in SINGLE 1% steps OR 10% steps

The first three bargraphs starting on the left hand side are LOW , HIGH and the third is (ABS DEV / Lo DEV / Hi DEV) Alarms, Blue when Healthy, Flashing Red when in Alarm and Yellow when Acknowledged

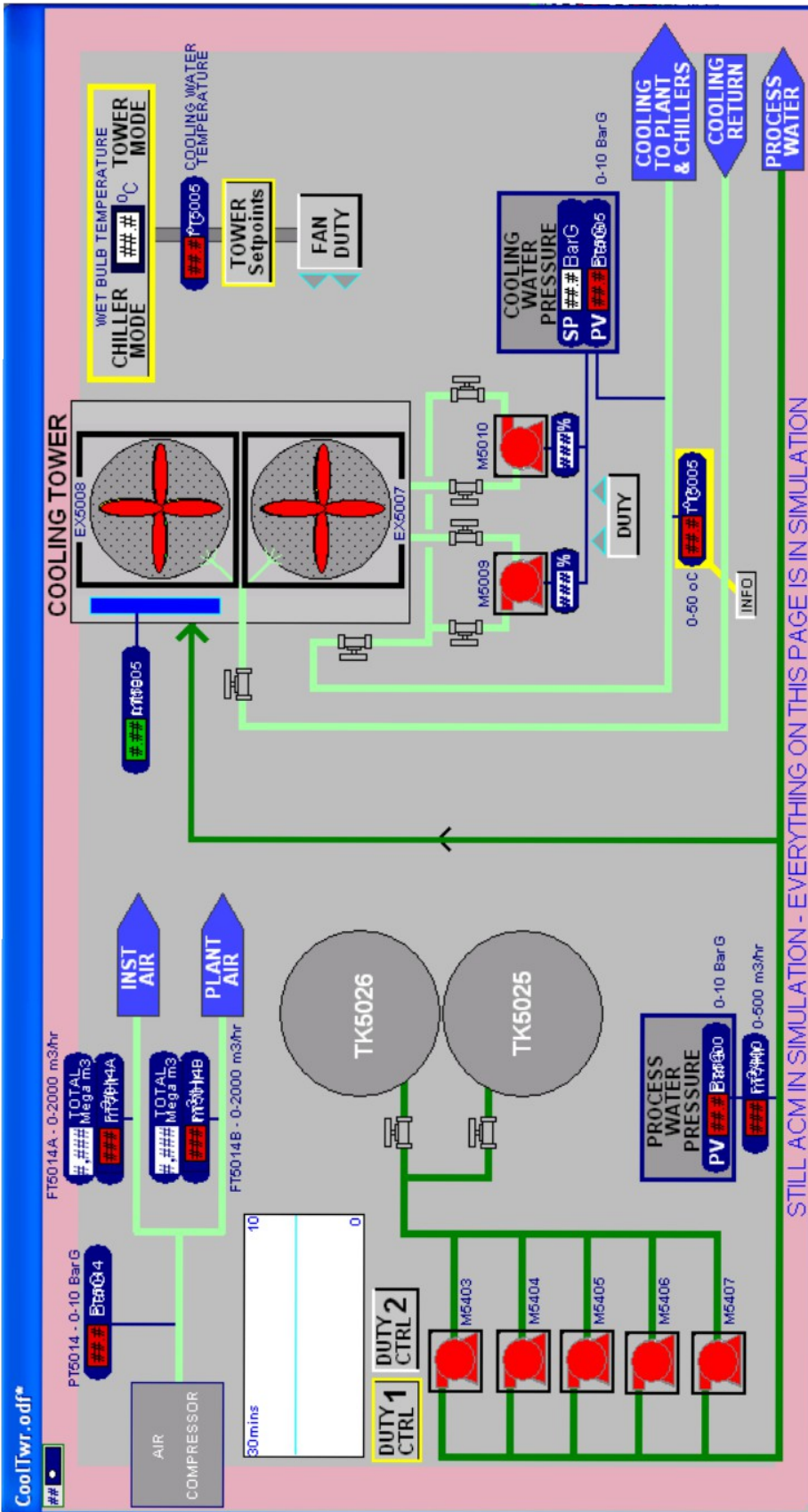


This analogue alarm block shows the PV and up to Four alarms for the PV, the little grey dots just before the DATADATA allow an engineer to Enable or Disable a particular alarm.

The alarm types are :-
High High , High , Low , Low Low
Abs Dev , Hi Dev and Lo Dev
So you could set up in 4mation all four alarms to be deviation alarms if you want.
The alarms are also Greyed out if they are not currently in use – just as when starting a pump.
This faceplate also has hidden alarm bargraphs to the left hand side just as the Controller faceplate.

Limitations of 4mation \ Benefits of S7

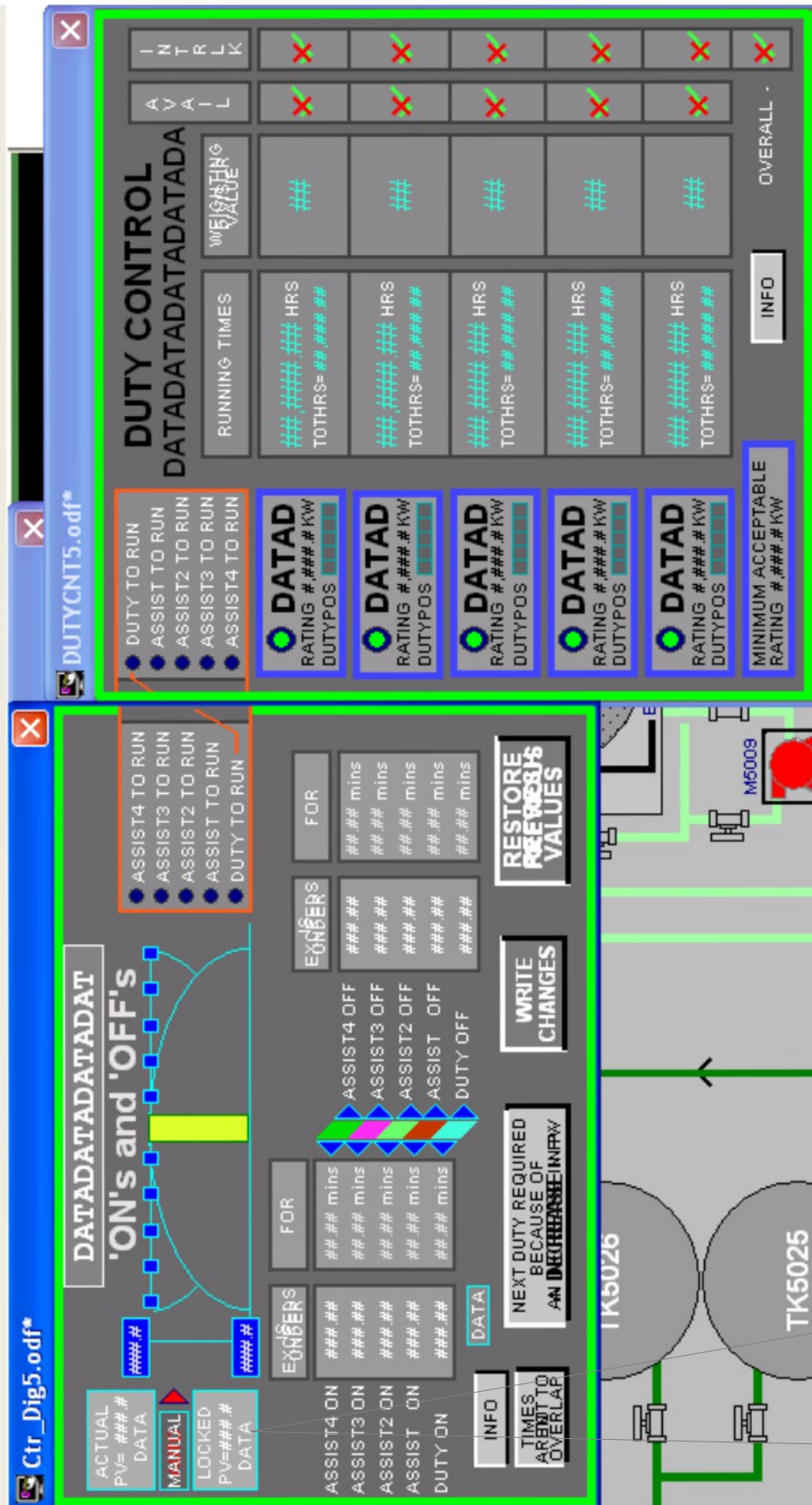
- S7 has a Cross Reference which is easier to use than APACS's FIND function
- in S7 when you are controlling variables – with Variable Monitor – you can drive more than one item with a single key press (in APACS you would have to use a MOVE block which affects the Time Stamp of the sheet) - but seeing as S7 doesn't have a Disable facility and you always have to change the Time Stamp of the function block to force something – means that the multiple forcing advantage of S7 is over-shadowed by the time stamp update issue.
- Printing out – like customers traditionally wanted – the printed version is very difficult to follow (because it is a graphical system).
- Can't think of any others ...



The Pink border shows that the 4mation controller is in simulation. Transferring from the Offline Simulation to the Online Real World is just as case of switching OFF a global variable !SimSTILL! (and this pink border would disappear)

All PVs have their units OR their Tag reference showing – you can just press F4 to toggle between them.

The following page shows the two faceplates that control the cooling water pumps.



There are five pumps to pump the cooling water to the tanks and here are the two faceplates that do that control. The Output of one faceplate becomes the input to the other.

This is exactly replicating what is going on in 4mation with an **array** of bits being the information that is passed from the Ctr_DigXX controller to the DutyControllerXX.

Where XX is 10 to never need to generate another Digi Controller OR Duty control block.

The Duty Control block has Weightings and Ratings in KWhr – just to confuse things some of the pumps are smaller than others and can't run on their own. But are still handled within the same block.

The INFO button brings up a TXT file which explains bits about the control of the faceplate.

There is a manual mode for this controller as well – you can disable the incoming PV and OVERWRITE the PV.

And the controller can cope with “an increasing PV requiring another pump to start” OR “a decreasing PV requiring another pump to start”

There is an option to allow overlapping of trip points.

I must apologise about showing you Fix32 graphics rather than 4mation coding – but in a hand-out like this it gives a better feel for what the software at the bottom end is doing. And by understanding these graphics you realise what is going on at the bottom end.

I have also coded a few other sites using 4mation including SLUVAD - a Welsh Water Treatment works in South Wales – with filter queues etc... 12 filters and a wash management system in each ACM – with two banks of 12 filters making up the whole of the site. (with a HART instrumentation network that gave a lot of grief).

Hope this oversight has been useful – let me know if you need any further advice – Greg McCormack – mccormackg@cougar-automation.com