Chapter 3

Foxboro DCS

3.1 Distributed Systems

The Intelligent Automation (I/A) Series system is an Open Industrial System (OIS) that integrates and automates manufacturing operations. It is an expandable distributed system that allows a plant to incrementally tailor the system to its processing requirements. The modules that make up the I/A Series system communicate with each other even though they can be located in a variety of locations. These locations depend upon the conditions and layout of the particular process plant. Another advantage of a distributed system is that each module has specific responsibilities. Each module independently performs its function regardless of the status of other modules [6].

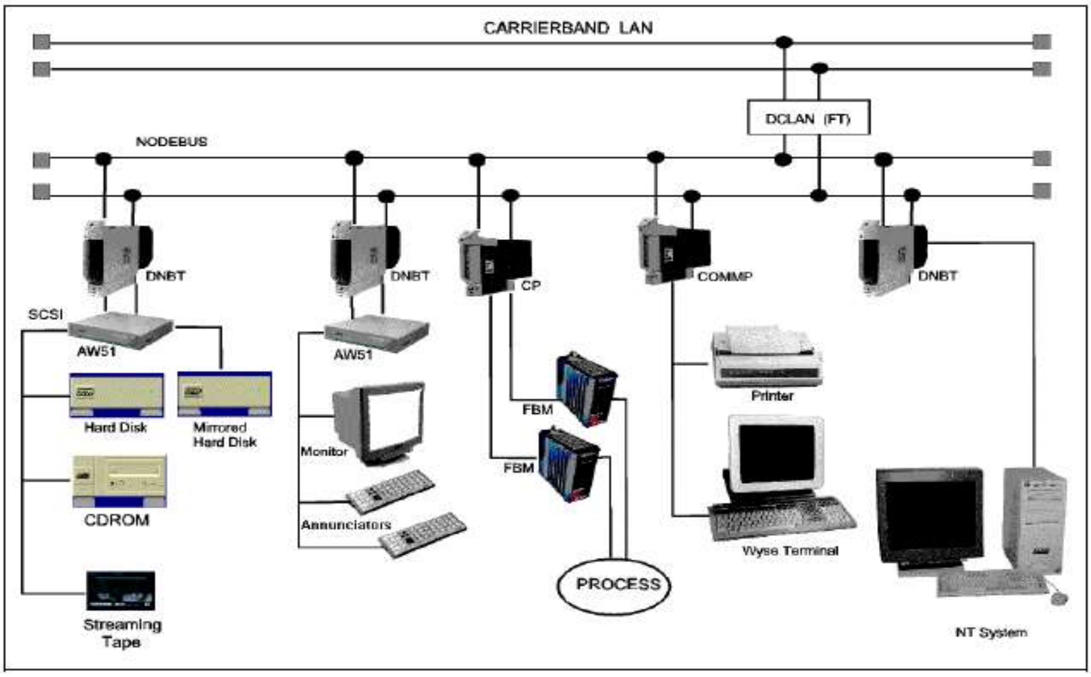


Figure 3-1: Distributed System

3.2 Hardware Overview

3.2.1 Modules Types

The I/A Series system is a set of devices, known as modules. Each module is programmed to perform specific tasks associated with the monitoring and controlling of manufacturing processes [6].

In order to meet specific needs at a plant, process control personnel create or modify software that executes within the modules. Various types of software and hardware configurations are better understood and successfully completed when personnel first become knowledgeable of the functions that each module performs [7].

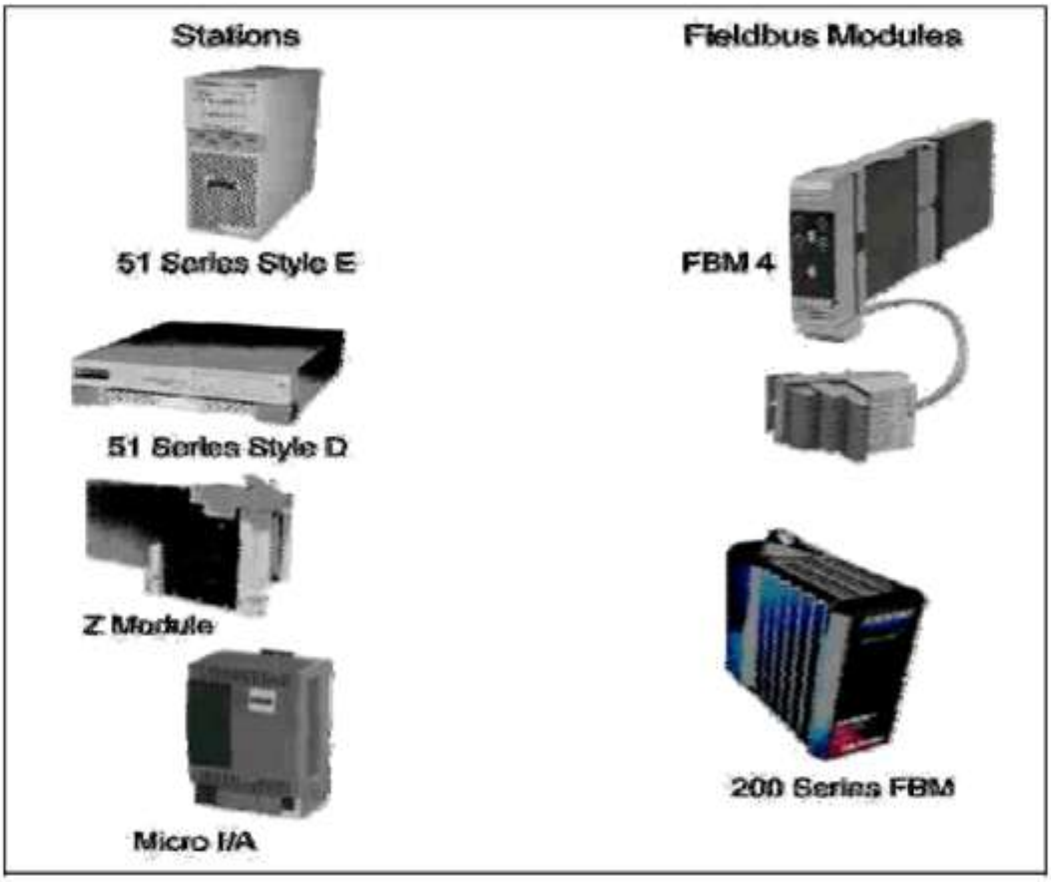


Figure 3-2: Module Types

3.2.2 Application Processor

The Application Processor (AP) performs computation functions. It contains the processing capability to run Foxboro-written system software, application packages, user-written programs and 3rd-party software (Historian) In addition, the AP acts as a file server or host to other stations on the network [6].

3.2.3 Workstation Processor

The main function of the Workstation Processor (WP) is to act as the human interface between the user and the process by providing the programs necessary to operate the user-interface. The WP is a graphics terminal from which operators interact with I/A Series displays to monitor and control process variables, receive notification of process alarms, take action on Process alarms, monitor system hardware health, display historized process data.

Process control personnel use the WP monitor to interact with I/A Series displays, operate the I/A Series configurators, and create and execute programs and reports [6].

3.2.4 Application Workstation

The Application Workstation (AW) is a combination of an AP and WP. Therefore, it can perform such AP functions as historical data collection and control database downloading. It also acts as a graphical interface similar to the WP. All the configurators can be accessed through this station. Again, there are different AW versions [6].

3.2.5 Control Processor

As the name implies, the function of a Control Processor (CP) is to automatically control or monitor the manufacturing process, or at least a portion of it. Upon boot up, the CP receives its control database from its host AP and stores it in its memory. A control database is a set of mathematical computations (algorithms) that perform specific functions, such as activating and deactivating motors and controlling valves. This control software is available through the Integrated Control Configurator (ICC) and usually modified by process control engineers. A (CP) communicates with the process through a set of Fieldbus Modules (FBMs) assigned to the CP by process control engineers. Again, keep in mind that there are different CP versions. The one that interfaces with the new I/A 200 Series family of FBMs is the CP60. It communicates with the FBMs via Fieldbus Communications Modules (FCMs). The Control Processor (CP) provides the functions necessary to communicate with other I/A Series stations and the process (via Field bus and FBMs). It executes control schemes tailored to the needs of a given process and control methodology [6].

A CP communicates directly with its fieldbus and can accommodate three control domains. The continuous control domain is used to implement feedback and feed forward control strategies, and monitoring. This involves analog and digital control blocks. The other two control domains are Ladder Logic, which involves programmable logic blocks (PLB) and sequence blocks, which provide user-defined sequence logic [6].

3.2.6 Field bus Modules

A Control Processor (CP) controls a process using its control database, which sends signals to and receives signals from the process (field devices) through a set of Fieldbus Modules (FBMs). But FBMs are NOT stations. Physically, they are smaller than stations and, logically, are on a separate communications path, called the Fieldbus [6].

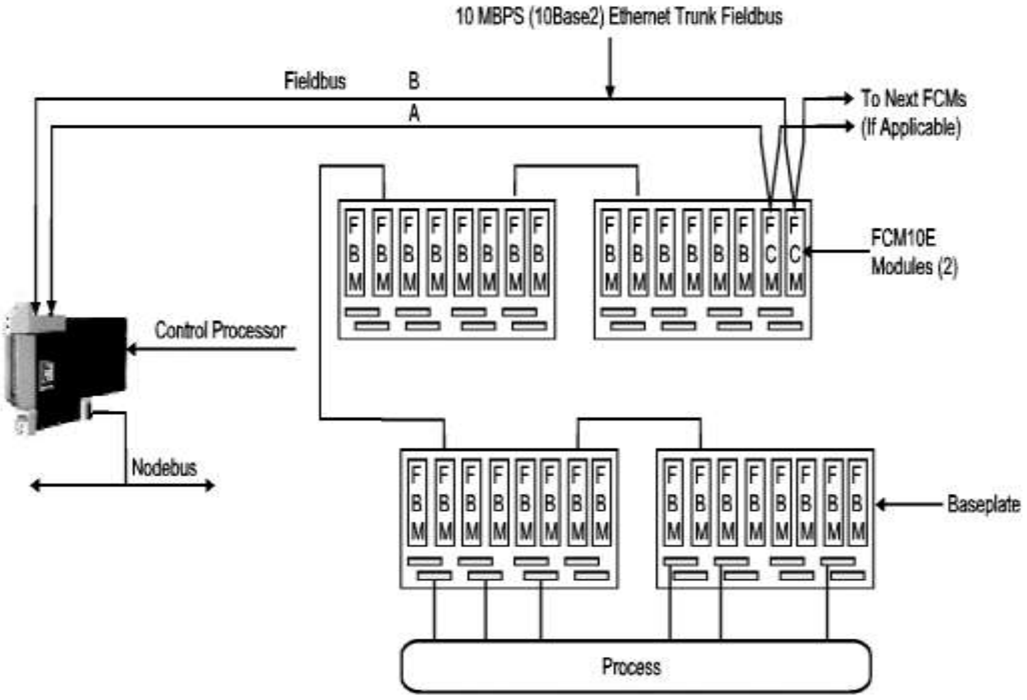


Figure 3-3: Control Processor and Field Bus Module Connection

FBMs connect directly to process monitoring and controlling devices, such as pumps, valves, transmitters and thermocouples. They pass information about the process to the CP where it is made available to the control strategy and to other stations on the I/A Series network. The purpose of an FBM is to modify a signal, making it understandable to the hardware receiving it. For example, an analog signal coming in from the field (process) must be converted to a digitalized signal in order for the CP to understand it. Conversely, a digitalized signal going out to the field needs to be converted to an analog signal. An FBM falls into one of two major categories analog and discrete (digital; on – off), these are lights showing the on/off state of each channel (signal). An FBM communicates with one and only one station, the CP to which it has been assigned. An FBM is directly connected to the process (pumps, valves, etc.). It passes information about the process to the CP, where the information is made available to the other stations on the I/A Series network. The CP also passes information to the appropriate FBM, which forwards it to the process for control purposes [6].

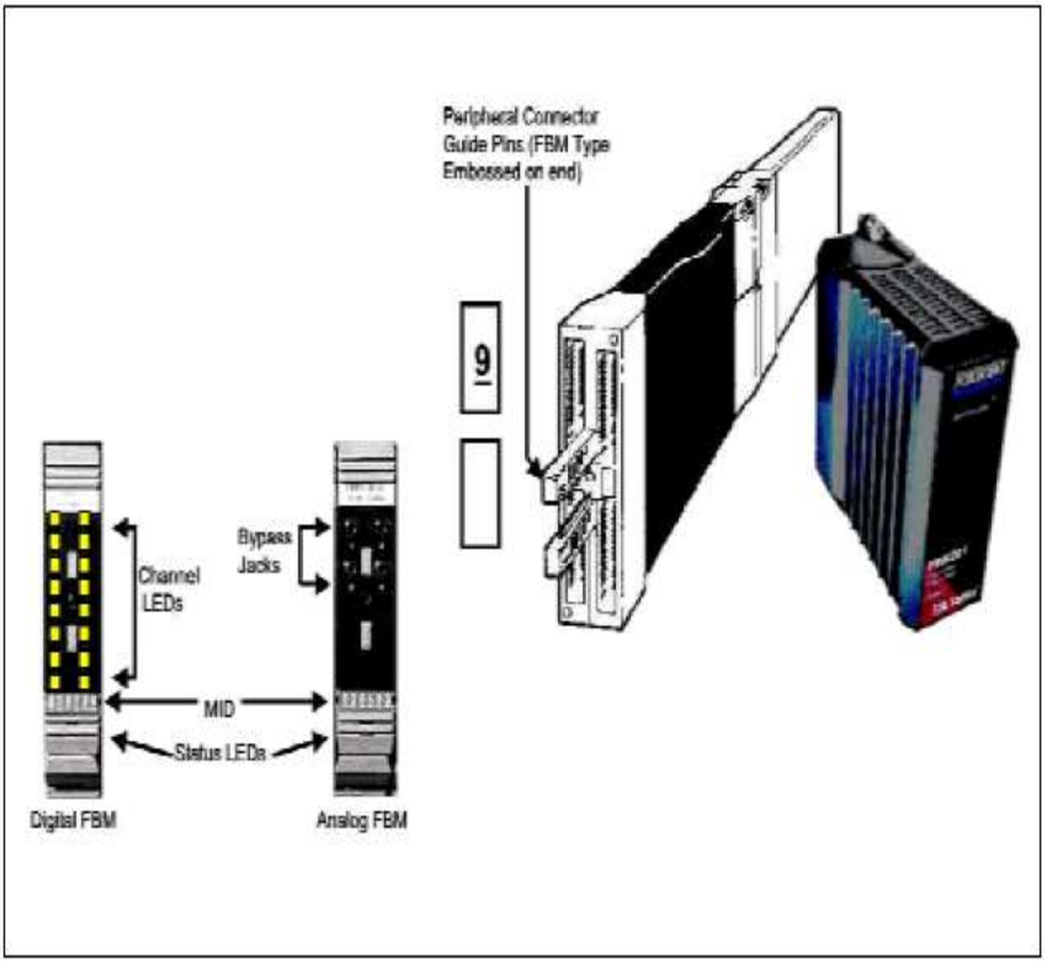
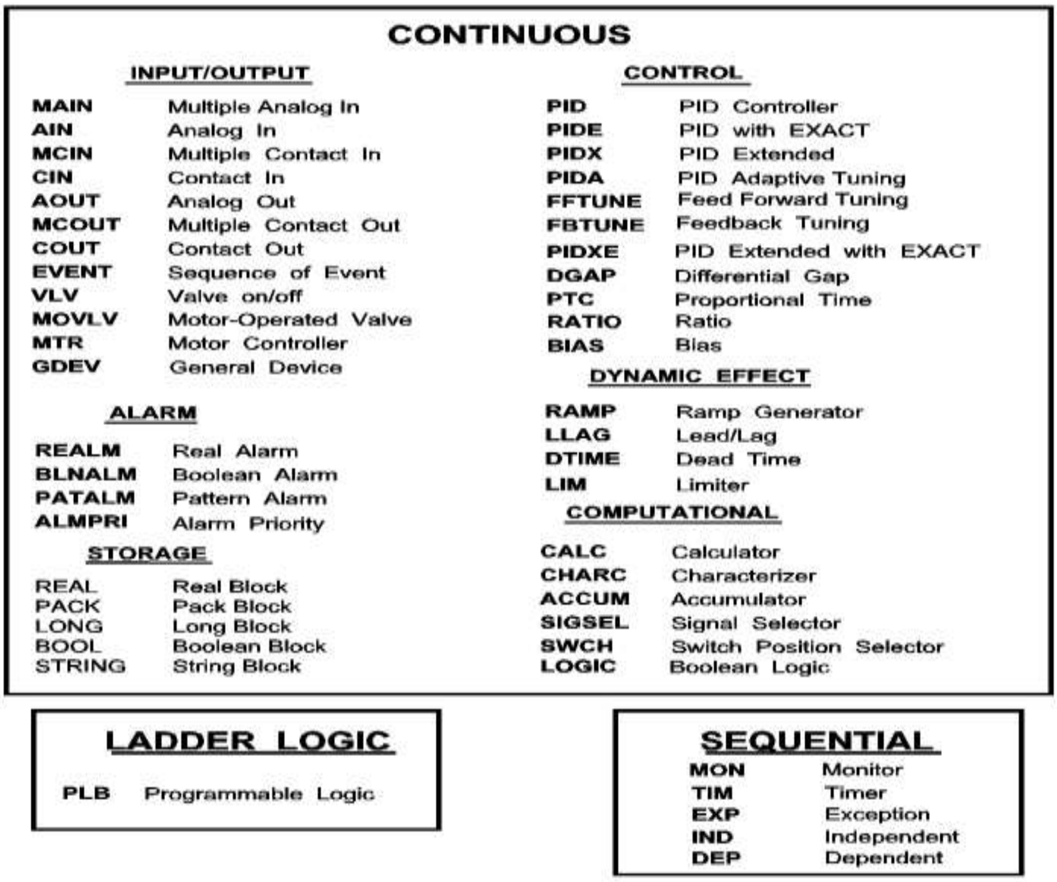


Figure 3-4: Field Bus Module

3.3 Software Overview

3.3.1 Integrated Control Configurator (ICC)

The Integrated Control Configurator (ICC) is intended for use by process control and applications Engineers. It provides procedures for configuring your process control database including Fieldbus Modules, equipment control blocks, compounds, and blocks and their parameters. The purpose of the Integrated Control Configurator (ICC) and I/A Series Configuration Component (IACC) is to create or modify process control databases. I/A Series software provides for the configuration of three types of process control schemes, referred to as domains continuous, Ladder Logic, sequence Logic Control [7].



CONTINUOUS

INPUT/OUTPUT

MAIN	Multiple Analog In
AIN	Analog In
MCIN	Multiple Contact In
CIN	Contact In
AOUT	Analog Out
MCOUT	Multiple Contact Out
COUT	Contact Out
EVENT	Sequence of Event
VLV	Valve on/off
MOVLV	Motor-Operated Valve
MTR	Motor Controller
GDEV	General Device

ALARM

REALM	Real Alarm
BLNALM	Boolean Alarm
PATALM	Pattern Alarm
ALMPRI	Alarm Priority

STORAGE

REAL	Real Block
PACK	Pack Block
LONG	Long Block
BOOL	Boolean Block
STRING	String Block

CONTROL

PID	PID Controller
PIDE	PID with EXACT
PIDX	PID Extended
PIDA	PID Adaptive Tuning
FFTUNE	Feed Forward Tuning
FBTUNE	Feedback Tuning
PIDXE	PID Extended with EXACT
DGAP	Differential Gap
PTC	Proportional Time
RATIO	Ratio
BIAS	Bias

DYNAMIC EFFECT

RAMP	Ramp Generator
LLAG	Lead/Lag
DTIME	Dead Time
LIM	Limiter

COMPUTATIONAL

CALC	Calculator
CHARC	Characterizer
ACCUM	Accumulator
SIGSEL	Signal Selector
SWCH	Switch Position Selector
LOGIC	Boolean Logic

LADDER LOGIC

PLB	Programmable Logic

SEQUENTIAL

MON	Monitor
TIM	Timer
EXP	Exception
IND	Independent
DEP	Dependent

Figure 3-5: I/Series Control Domains/Blocks

Within these domains, standard algorithms (blocks) are available to create specific process control applications. These blocks are not restricted to any particular control domain and can be utilized for any application. Blocks in one domain can be configured to interact with blocks in another. Use of any combination of blocks to control a process is the product of the engineer's knowledge, skill and imagination [6].

3.3.1.1 Continuous Control Domain

A continuous (AIN, PID, AOUT) control block periodically receives and/or sends an updated process value. Process values relate to measurable physical conditions such as flow, liquid level, pressure, temperature, and Ph [6].

3.3.1.2 Ladder Logic Control Domain

This domain represents relay logic that is connected in parallel and series paths to operate equipment in a predefined manner. Ladder Logic provides on/off control, which is faster than control programs running in the control processor. Ladder logic software can run in any discrete Field bus Module, with the programmable logic block (PLB) acting as the interface between the ladder logic and the CP. It is the responsibility of the PLB Block to provide a dedicated interface between a ladder logic scheme running in a user specified FBM and the control schemes running within CPs [6].

3.3.1.3 Sequence Logic Control Domain

This domain performs programmable logic control, using the High-Level Batch Language (HLBL). Sequence logic can interface and interact with the continuous and ladder logic domains. The five blocks in this domain are the MON, TIM, EXC, DEP, and IND (Independent) block acts as a supervisory block to control the overall process [6].

3.3.2 Blocks and Compounds Concept

Blocks that are utilized to control a loop or a part of the process are chosen to be members of a particular group, known as a compound, which resides in a CP. Usually, more than the one compound resides in a CP and thus defines the control database for that CP. Two simple control strategies, a continuous control scheme and a ladder logic scheme, are depicted as examples in the next two figures [6].

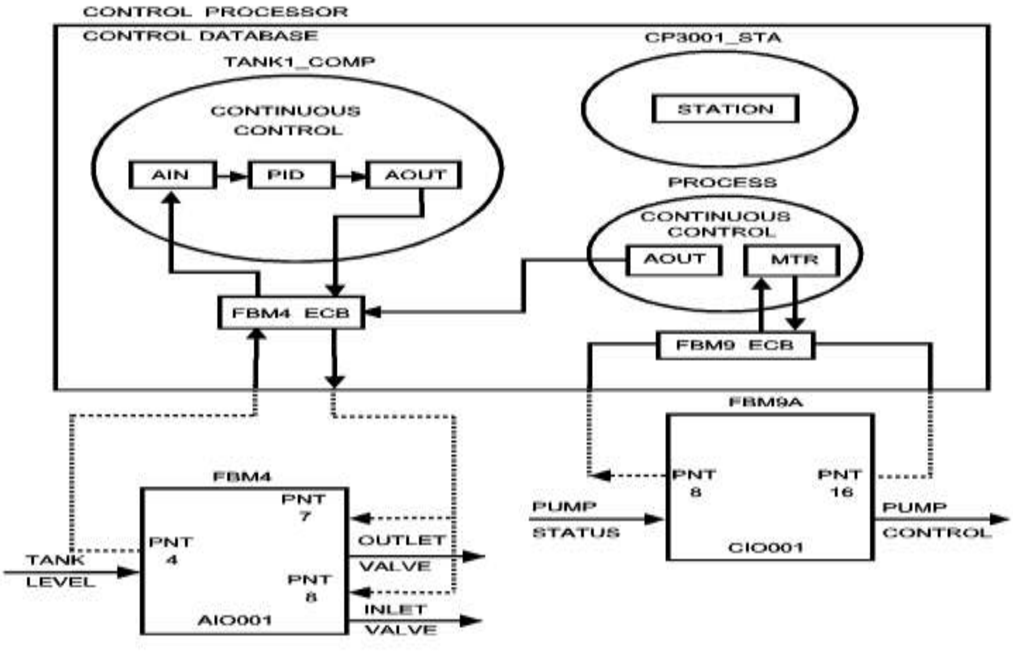


Figure 3-6: Compound Concept

3.3.2.1 CALCA – Advanced Calculator Block

The Advanced Calculator (CALCA) block provides both logical functions and arithmetic computational capability within one integrated environment. This block provides dual-operand efficiency in several mathematical and logical instructions. The configuration process allows you to program the block by entering a series of up to 50 programming steps. Each program step is represented by a parameter string of up to 16 characters [7].

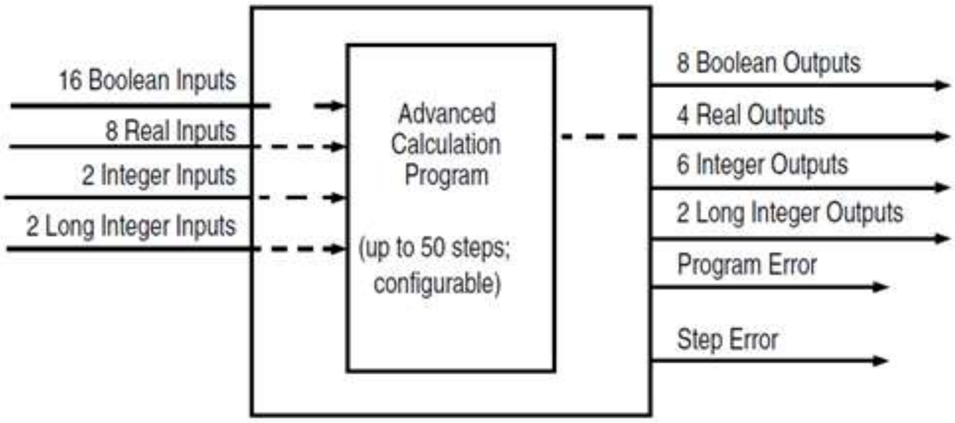


Figure 3-7: Advanced Calculator Block

The CALCA block operates like most programmable pocket calculators. The block's input/output modularity and programming capacity provides a level of functionality that complements the standard block set offering. The program which you enter is executed once each time the CALCA block executes. A single execution of the program is defined as a single consecutive execution of each program step in turn. It is, however, possible to specify conditional execution of steps or groups of steps. Every program step contains an opcode, which identifies the operation to be performed, and up to two command line arguments. The command line arguments consist of the actual operands for the step, the location of the operands, a specification of details which further refine the opcode, or some combination of these factors. The CALCA block can operate in Auto or Manual mode. The operation of the block in Manual is identical to the operation in Auto, except that any output parameters involved in the steps are not modified by the block logic. The bock features are Provides 8 real inputs, 2 long integer inputs, 2 integer inputs, 16 Boolean inputs, 4real outputs, 2 long integer outputs, 8 Boolean outputs, and 6 integer outputs. In addition Provides 24 floating point memory data storage registers that are preserved between execution cycles, provides 50 programming steps of up to 16 characters, allowing dual operands in all appropriate instructions, supports Auto/Manual capability for all block outputs; in Manual, all functions are performed normally except for changes of output values, allows forward branching of program control, however, backward branching is not allowed, to prevent endless loops and Lets you initialize all timers and memory registers [7].

The CALCA bock functions are provides 114 program instructions, consisting of 32 Arithmetic instructions, which perform mathematical calculations, 13 Boolean instructions, which provide various Boolean and packed Boolean operations, 27 Input/output instructions, which perform the input or output of data or status bits, 2 Cascade and Propagation instructions for error propagation and downstream cascade acknowledgments, 10 Memory and Stack Reference instructions for reading and writing the memory registers, or directly manipulating the stack, 10 Program Control instructions for conditional and unconditional branching, and program termination, 9 Clear/Set instructions, which provide for conditional or unconditional clearing or setting of parameters, sometimes with simultaneous program control action, 6 Timing instructions which control the operation of program timers, 2 Logic instructions which emulate the operation of flip-flops, 3 Error Control instructions which provide access to the run-time error flag [7].

3.3.2.2 AIN – Analog Input Block

The Analog Input Block (AIN) receives an input value from a single point of an analog, pulse count, or Intelligent Field Device type of Fieldbus Module (FBM) or Fieldbus Card (FBC), or from another block, and converts it into suitable form for use in an I/A Series control strategy. The input to the block is the digitized data from the FBM's Equipment Control Block (ECB), and the output is a value expressed in the appropriate engineering units [7].

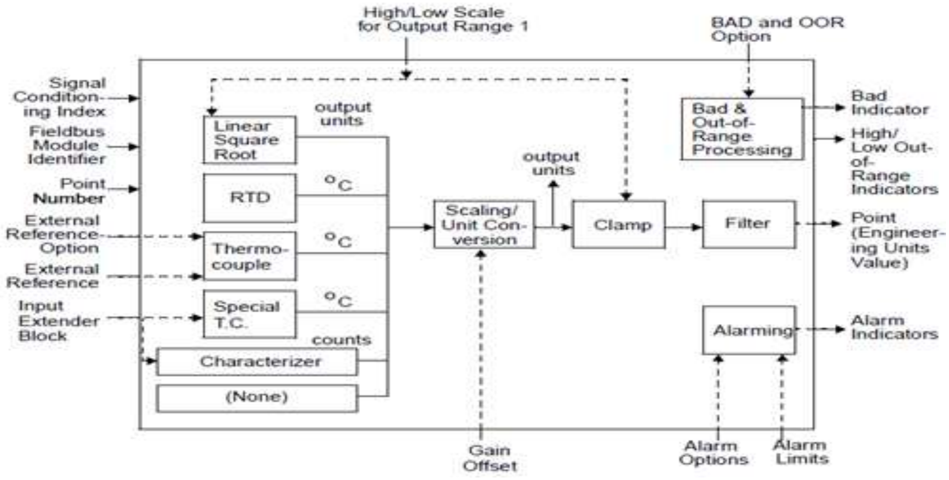


Figure 3-8: AIN – Analog Input Block

The AIN block interfaces to an ECB which stores values from an FBM or FBC receiving analog electrical signals, pulse inputs, or pulse rate inputs. Each execution cycle, the FBM or FBC transmits the values to its associated ECB in the Control Processor, and the AIN block reads its operational status, the channel status, and the input data from the ECB for the specified point. After validation, the AIN block converts the channel's raw data to a floating point number in engineering units, according to the configured Signal Conditioning Index, the High and Low Scale of the engineering units range, and any specified units conversion or filtering option. This output is presented to the control strategy as Point (PNT). The original value as present in the ECB is preserved and made available as output Raw Count (RAWC).The sequence of signal processing actions is, in the order of application: Signal Conditioning or Characterization, Scaling for Unit Conversion, Clamping, and Filtering. Different alarms can be configured like High/Low Absolute alarm, High-High and Low-Low Absolute alarm, Bad point alarm [7].

3.3.2.3 CIN – Contact Input Block

The Contact Input Block (CIN) provides the control strategy with input capability for a single digital input or digital output point in any Fieldbus Module (FBM) or Fieldbus Card (FBC) containing such points. The block also provides alarm handling features relating to the input [7].

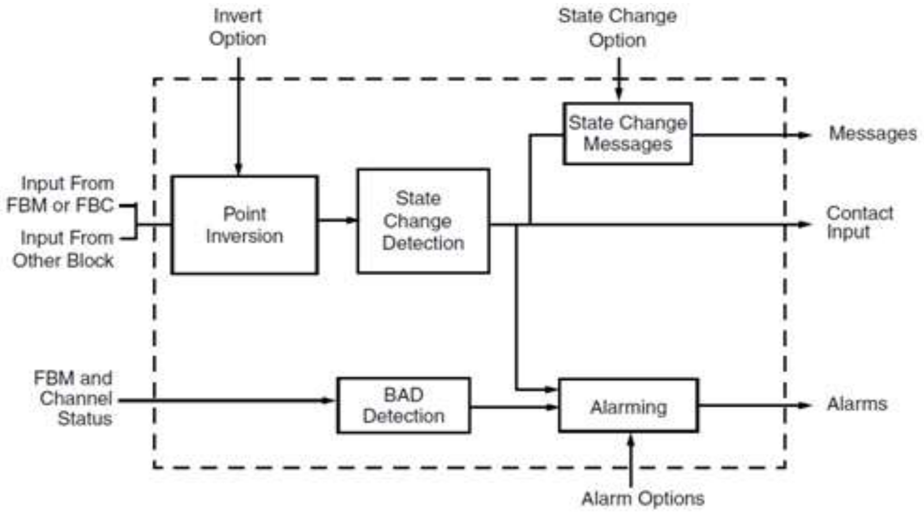


Figure 3-9: CIN – Contact Input Block

The CIN block interfaces to an Equipment Control Block (ECB) which stores digital input or output values from an FBM or FBC. Each execution cycle, the block presents the value of the specified digital point at its output, called Contact Input (CIN). When no FBM or FBC is configured, the block input is taken from another block connected to the Input (IN) parameter. The CIN block provides optional point inversion, Bad I/O and State alarming, State Change message processing, and Auto/Manual capability [7].

3.3.2.4 COUT – Contact Output Block

The Contact Output block (COUT) provides the control strategy with output capability for a single digital output point in any Fieldbus Module (FBM) or Fieldbus Card (FBC) containing such points. You can select a sustained output that follows the block input or a pulsed output with a selectable pulse width. The block also supports output inversion and alarming[7].

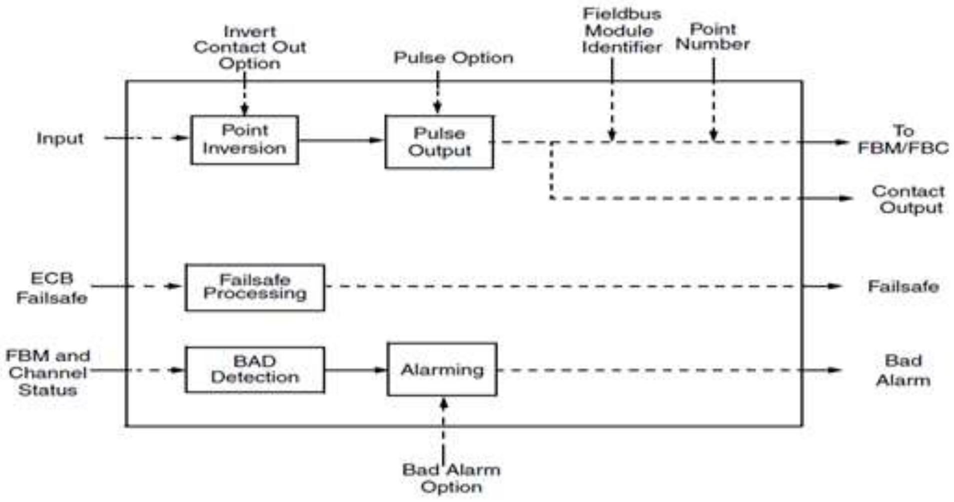


Figure 3-10: COUT – Contact Output Block

The COUT block interfaces to an Equipment Control Block (ECB) which contains digital output values to an FBM or FBC. Each execution cycle, the block writes the value of the input parameter Input (IN) to its output, called Contact Output (COUT). When an FBM or FBC is configured, the value of COUT also drives the connected output point. The COUT block provides optional point inversion under control of parameter Invert Contact Output (INVCO) and optional pulsed output under control of parameter Pulse Option (PLSOPT). The block uses parameters Initialize out (INITO) and Failsafe (FS) to alert upstream blocks to various abnormal situations. Bad I/O alarming and Auto/Manual capability are also provided [7].

3.3.2.5 ACCUM – Accumulator Block

The Accumulator block, ACCUM, provides a convenient method for accumulating values from an analog flow signal source, without using a pulse input primary device. The block accepts a real input, as a rate or pulse count, and then scales it to create the correct rate units. ACCUM then integrates the input over the block period time, accumulates the value with the results of the previous block period time(s) and stores it as a real value to the output. Scaling is accomplished using a meter factor to achieve engineering unit and time scale compatibility between the measurement inputs and the desired output units [7].

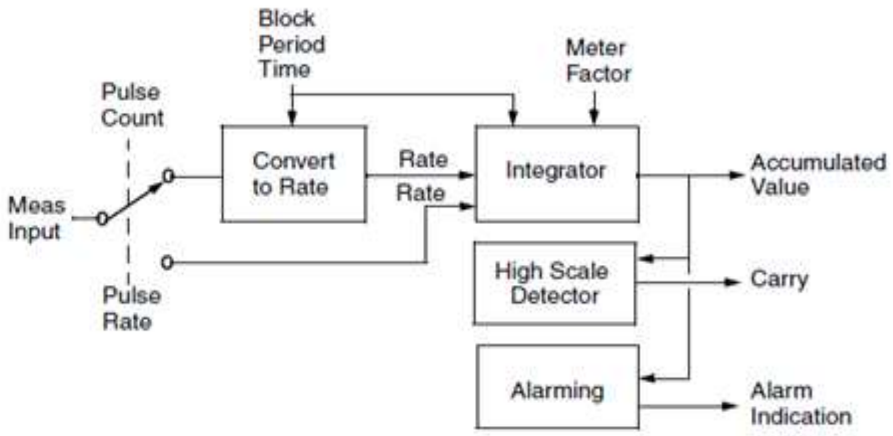


Figure 3-11: ACCUM – Accumulator Block

The ACCUM block integrates the measurement input over the block’s period time, adds it to a running total for the block, and then saves it to the output. An optional meter factor can be used to make the output units compatible with desired output units. In all cases the meter factor expresses a flow volume per pulse. The pulse count option allows you to configure the type of the pulse count option allows you to configure the type of totalization performed and determines whether totalization is block period time dependent or independent [7].

3.3.3 Human Machine Interface

The term Human Interface refers to the way a workstation is set up to interact with the user.

3.3.3.1 Fox View

Many I/A Series systems use Fox View as its interface. Fox View allows a user to utilize numerous applications to respond to alarms, collect and interpret data, modify process variables, perform on-line trending, and generate reports.

The Human Interface may differ from one location to another, but there are common features that are discussed below. The FoxView interface includes a display Area, menu, system, display, and status Bars [6].

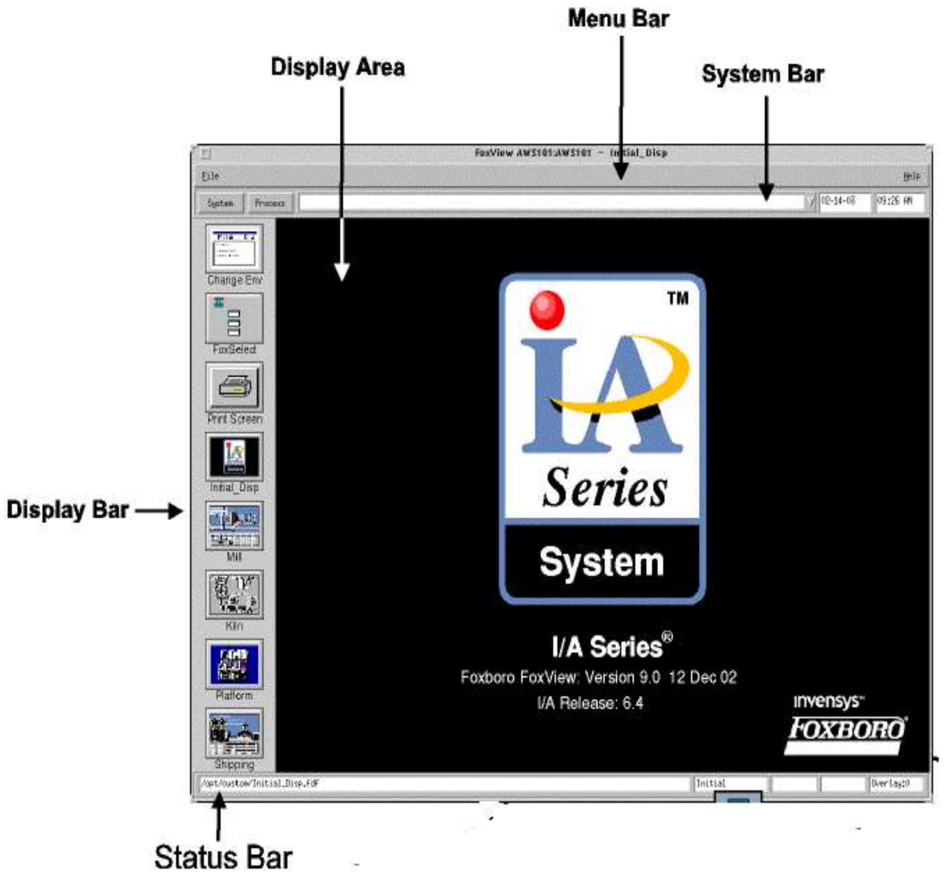


Figure 3-12: I/A Series FoxView Human Interface

3.3.3.2 Fox Draw

The Fox Draw™ software application is a graphical display editor for creating and maintaining displays for viewing process control dynamics. A display is a file that is constructed and configured to be viewed from a Fox View™ window or a Display Manager window. A display can represent a plant, a process area, or a detailed portion of a process. A display can be configured to allow operator interaction with the process by moving objects or typing inputs. A display is composed of objects, each of which can be configured with attributes. Object attributes determine the object’s static and dynamic appearance and the actions an operator can perform on an object [6].

The term “object” includes primitive objects (such as lines, rectangles, circles, and text), library objects, and bitmaps. Library objects include Invensys Foxboro supplied and user-built symbols, overlays, faceplates and trends. The term “symbols” refers to the vast collection of objects that you can copy to a display from Invensys Foxboro supplied and user-build palettes. The Fox Draw application is integrated into I/A Series Configuration Component (IACC) software as an application that reads and writes common data to the IACC database. The IACC Fox Draw Editor supports all the functionality of a stand-alone Fox Draw Editor. The interface requires that the Fox Draw software be installed on the same workstation as the IACC client application. The Foxboro display file is represented by a reference object in the IACC database [6].

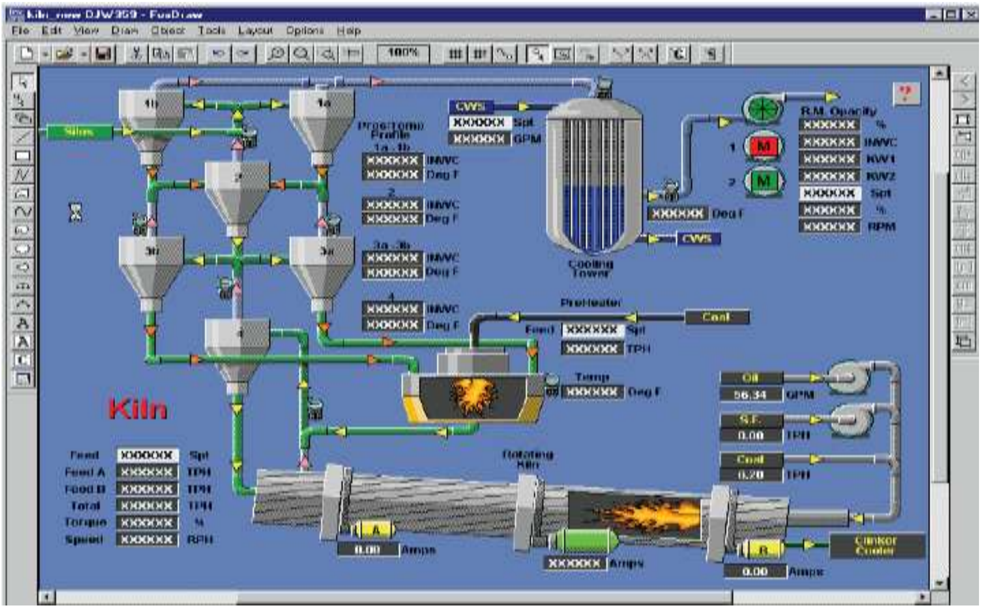
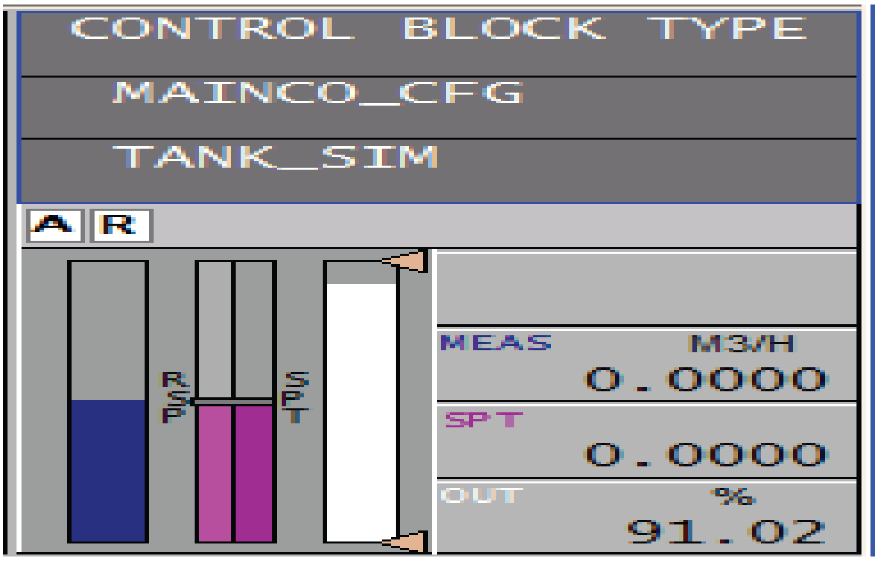


Figure 3-13: Fox Draw

3.3.3.3 Faceplates

Faceplates include supplied faceplates and user-built faceplates. Faceplates are library objects and have an .m1 extension. You can build a base display that contains up to eight faceplates [6].


CONTROL BLOCK TYPE

MAINCO_CFG

TANK_SIM

A R

MEAS M3/H 0.0000

SPT 0.0000

OUT % 91.02

Figure 3-14: Faceplates

3.3.3.4 Trends

A trend can monitor the behavior of one or more process variables over time. You can configure up to eight trend graphs on a single base display (or base display plus library object). The data values being graphed can be from real time or historical data. FoxDraw provides many trend types from which to choose. Depending on trend configuration and operator permission (access levels) in FoxView, operators can reconfigure and save trends [7].

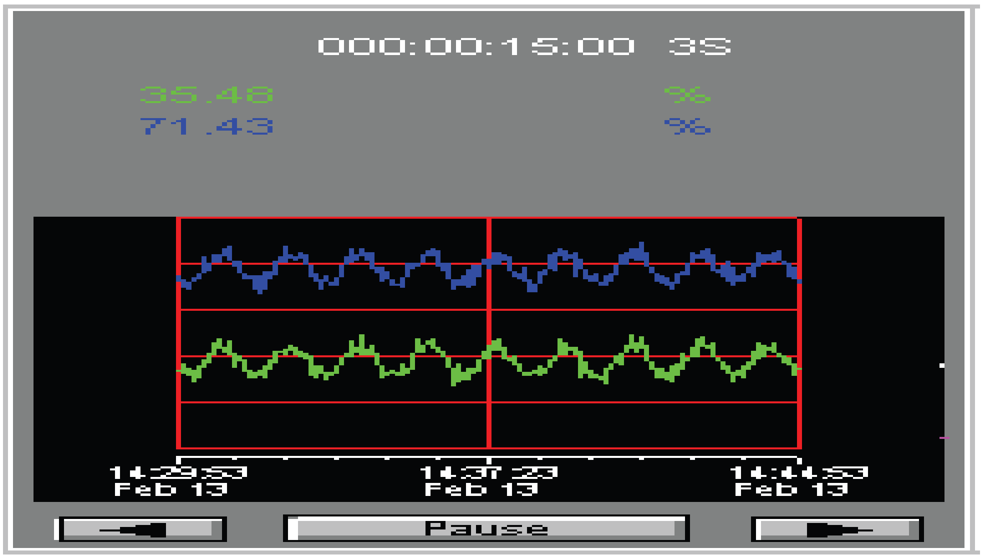


Figure 3-15: Trends

The trend display can include up to four user-configured horizontal lines to mark boundary values. In the configuration dialog boxes, these static lines are labeled HI%, HI-HI%, LO%, and LO-HI%. FoxView indicates status conditions by using different line styles. A dashed line indicates bad status [7].