

---

## Definition of Function Blocks

### Definition

A *function block* is a unit of software that performs a set of operations on its *input parameters* and *function block parameters* and produces *output parameters*. These output parameters can be programmed as inputs to other function blocks, whose output parameters can be programmed as inputs to other function blocks, and so on. By programming all desired function blocks' input parameters and function block parameters, you configure the instrument to measure and control your process.

### Types of function blocks

Each function block performs a set of operations which fulfills a unique purpose. For example, the Analog Input function blocks processes the analog input data, the alarm function block processes alarms, and so on.

Table 3-3 describes each function block.

Some function blocks—namely, Analog Input, Analog Output, Discrete Input, and Discrete Output—interface with the hardware; that is, they are the link between the instrument and the input or output hardware. The Analog Input and Discrete Input function blocks convert the incoming process data (like the process variable or any discrete on/off signals from a switch) into information usable by the instrument. This incoming information is processed according to the entire function block configuration in the instrument, and it is ultimately passed on to the output function blocks. The Analog Output and Discrete Output function blocks convert this output information into a voltage or current which is fed to the corresponding output hardware (like a current output or relay).

Other function blocks are not directly “seen” by the hardware; they are purely software. They can be thought of as the middle of the process described in the previous paragraph. For example, a Standard Splitter Calculated Value can split a control loop's output into 2 values: one for heating and one for cooling. These 2 values can be passed on to the Analog Output function block which ultimately controls the amount of output current or voltage.

### Flow of information

The “flow” of information— from the input hardware to the input function blocks to the function block configuration to the output function blocks to the output hardware—can be likened to a river flowing from upstream to downstream. In some cases, like with a control loop's feedback, this analogy is not true because the information is flowing in a circle, but it is a helpful way to view how function blocks are generally interconnected. For example, the Analog Input function block is typically upstream of the Control Loop function, which is typically upstream of the Analog Output function block. Of course, if two function blocks are not directly or indirectly connected, there is no flow between the two. Just remember that every function block has input, does a set of operations, and produces an output. When several function blocks are linked together, there is a flow of information.

---

**Table 3-3 Function Block Types**

Function block name	Type	Maximum available*	Purpose
Alarm	AL	96	Causes alarms under specified conditions.
Analog Input	AI	48	Interfaces with measuring input hardware (thermocouple, RTD, mA, volts).
Analog Output	AO	8	Interfaces with analog output hardware (current output (CAT)) or with output relay hardware (time proportion (DAT)).
Calculated Value	CV	96	Performs various calculations on specified parameters.
Constant	CN	32	Outputs a number or an analog parameter value.
Discrete Input	DI	36	Interfaces with discrete input hardware (dry contact closure).
Discrete Output	DO	36	Interfaces with output relay hardware (AC relay, DC relay, mechanical relay, open collector output).
Loop	LP	8	PID or ON/OFF control with various outputs.
Setpoint Profiler	SP	4	Generates a time-varying setpoint for a loop's <b>Setpoint #2</b> .
System	SY	1	Outputs discrete status of alarms, data storage, and diagnostics; outputs analog value of reference junction temperature. This function block is not programmable; its outputs are produced automatically.
Totalizer	TL	48	Outputs accumulated total over time.
* Depends on options ordered.			

**Why use function blocks?**

Function blocks give you configuration flexibility. For instance, the instrument does not have a dedicated relay that is activated during an alarm; instead, you can program any of several Alarm function blocks to control any relay. Also, there is not a specific input for your process variable; any of several Analog Input function blocks can be programmed to be your process variable. In general, function blocks let you connect the output parameter of any function block to the input parameter of any function block.

---

## Components of function blocks

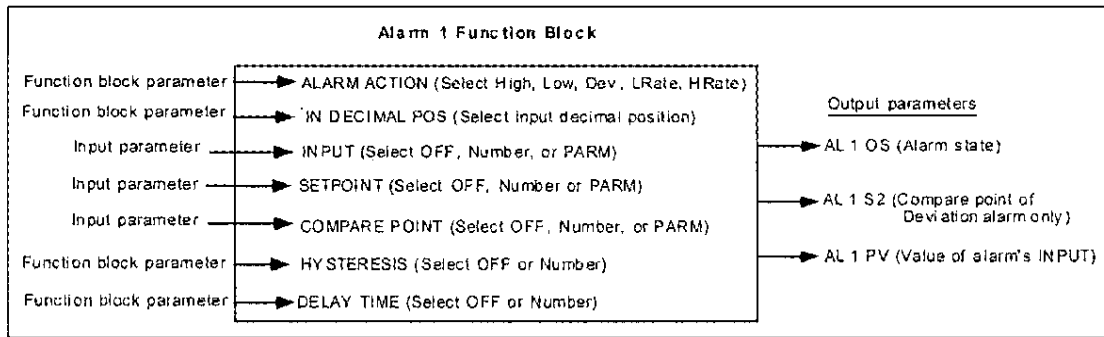
The three components of a function block are:

input parameter(s)

Function block parameter(s)

Output parameter(s).

Figure 3-7 shows the function block Alarm #1's components.



**Figure 3-7 Alarm 1 Function Block Components**

### Input parameter

A function block's input parameter can be configured to be OFF, a number, or it can receive its data from outside the block from another block's output codes. These output codes are shown in Table 3-4. That is, an input parameter is any menu item that can be programmed as (connected to) one of these output codes. These output codes are grouped under the menu choice PARM. When you are programming a function block and one of your choices is PARM, you know you are programming an input parameter. See Figure 3-7.

For example, suppose you are programming an alarm function block. One of the alarm's menu items is INPUT, which specifies which point will be monitored for an alarm condition. One of the choices for the INPUT is PARM, which lets you connect the INPUT to one of the output codes in Table 3-4. Therefore, the INPUT is an input parameter because it receives its data from another function block.

Some function blocks can have multiple input parameters. For example, an Alarm function block has an INPUT and a SETPOINT, both of which can be connected to other function blocks.

Discrete Input function blocks have no input parameters; that is, they have no inputs that can be connected to another block's output codes.

---

---

## Function block parameter

A function block parameter's data is contained within the block. When you are programming a function block and are not given a choice of PARM, you are programming a function block parameter. Typical choices when programming a function block parameter are NONE, OFF, any numerical value, or a list of options for the parameter, but not PARM. See Figure 3-7.

For example, to program an Alarm function block's ALARM ACTION, you select from a list of choices: NONE, LOW, HIGH, DEV, LRATE, HRATE.

Other function block parameters are an Analog Input's RANGE LOW and RANGE HIGH, where you specify the voltage range or temperature range.

## Output code

An output code is the result of the function block's operations on the input parameters and function block parameters. It is designated by one of the two-character output codes shown in Table 3-4. An output code can be programmed to be the input to one or several other function blocks. See Figure 3-7.

Output codes are either discrete (can be on or off) or analog (numerical value). For example, DI1 OS is the output status of Discrete Input #1: on or off. AI1 OV is the output value of Analog Input #1: a voltage or temperature. Therefore, a discrete input parameter must be programmed with only a discrete output code, and an analog input parameter must be programmed with only an analog output code.

---

## ATTENTION

The function block SYSTEM PARAMETER, abbreviated SY, does not have input parameters or function block parameters like the other function blocks; SY produces output codes only. These output codes, shown in Table 3-4, are mostly values or states that indicate the status of system-wide parameters. For example, if any Alarm function block's output status is ON, the SY function block's AG (alarm global) output code is also ON.

Another example is the SY F1 output code, which produces a quick ON-to-OFF discrete signal when the F1 key is pressed. This SY F1 can be used as a trigger to another action. **For example, to allow an operator to start the Profile or reset the Totalizer by pressing the F1 key, you can program a Setpoint Profile's Start parameter or a Totalizer's Reset parameter with SY F1.**

---

**Table 3-4 Function Block Parameter Designators**

Function Block Type	Function Block Name	Output code	Parameter Name	Parameter Type
AI	Analog Input	OV	Output Value	Analog
SY	System Parameter	RT	Reference Junction Temp.	Analog
		AG	Alarm Global	Discrete
		AH	Alarm High	Discrete
		AL	Alarm Low	Discrete
		DF	Diagnostic failure	Discrete
		DG	Diagnostic General	Discrete
		SF	Storage Full	Discrete
		SW	Storage Warning	Discrete
		AX	Analog Safe Parameter	Analog
		DX	Discrete Safe Parameter	Discrete
		F1	F1 or Start Key on keyboard	Discrete
		F2	F2 or Hold Key on keyboard	Discrete
		F3	F3 or Reset Key on keyboard	Discrete
		F4	F4	Discrete
F5	F5	Discrete		
AO	Analog Output	OV	Output Value	Analog
		BC	Back Calculation Value (Feedback)	Analog
		S2	Process Variable (AO's input)	Analog
DI	Discrete Input	OS	Output State	Discrete
DO	Discrete Output	OS	Output State	Discrete
LP	Control Loop	OV	Output Value	Analog
		PV	Process Variable	Analog
		DV	Deviation Value	Analog
		WS	Working Setpoint	Analog
		S1	Setpoint #1 Value	Analog
		S2	Setpoint #2 Value	Analog
		BC	Back Calculation Value (Cascade feedback)	Analog
		AM	Auto/Manual Status	Discrete
		SS	Setpoint #1/Setpoint #2 Status	Discrete
		ON/OFF Loop only	OS	Output Status

**Table 3-4 Function Block Parameter Designators (continued)**

Function Block Type	Function Block Name	Output code	Parameter Name	Parameter Type
SP	Setpoint Profiler	OV	Output Value	Analog
		A1	Auxiliary Output Value	Analog
		PV	Process Variable (Guaranteed Soak PV #1)	Analog
		SN	Segment Number	Analog
		SH	Hold Status	Discrete
		SE	End Status	Discrete
		SA	Active Status	Discrete
		Si	Active or Hold Status	Discrete
		SR	Ready Status	Discrete
		E1	Event#1 Output	Discrete
		thru	thru	
		E9	Event#9 Output	↓
		EA	Event#10 Output	
		EB	Event#11 Output	
thru	thru	Discrete		
EG	Event#16 Output			
AL	Alarm	PV	Process Variable (alarm's input)	Analog
		S2	Compare Point (of Deviation alarm)	Analog
		OS	Output Status	Discrete
CN	Constant	OV	Output Value	Analog
		PV	Process Variable (Constant's input)	Analog
CV	Calculated Value*	OV	Output Value	Analog
		PV**	Process Variable	Analog
		A(n)	Analog Output #n	Analog
		BC	Back Calculation	Analog
		S2	Auxiliary input (link to totalizer preset)	Analog
		D(n)	Discrete Output	Discrete
OS	Output Status	Discrete		
<p>*CV output codes are available for programming only if the CV has been programmed. For example, you cannot program an input parameter with CV1 OV unless CV1 has been programmed.</p> <p>**Input to the following CV types: Peak Pick, 1 Point Block Avg., 1 Point Rolling Avg., Scaling, Signal Select</p>				
TL	Totalizer	OV	Output Value	Analog
		PV	Process Variable (Totalizer's input)	Analog
		OS	Output Status	Discrete
		S2	Preset Value	Analog