MY ACCOUNT LOG IN

Login

Password

Ok

If you don't have a login, click here to register

Forgot your password?

Profibus-PA: status byte and Fail-Safe

Tweet G+ Share

César Cassiolato
Marketing, Quality, Project and Services Engineering Director
SMAR Industrial Automation
cesarcass@smar.com.br

Introduction

> A very useful feature provided by the Profibus PA protocol is the status byte in the float+status structure.
>
> The big advantage of this feature is to allow a view the measurement value and information that can qualify this value and to assist in safety fault condition (Fail-Safe). We will see below some details about this feature.

Understanding the status byte (state)

The input and output parameters status is intended to inform to other function blocks the current state.

The status is composed of three parts: Quality, Sub-state and Limits.

Quality – indicates the quality of the parameter value.

- *Good Cascade* – The value quality is good and can be used for control in cascade.
- *Good Non-cascade* –The value quality is good and cannot be used for control in cascade.
- *Uncertain* – The quality value is below normal, but value can still be useful.
- *Bad* – The value isnot useful.

Sub-status – The sub-status is a supplement of quality state and takes information to start or stop a control in cascade, alarms and others. There are different sub-status configurations for each quality.

Limits – Provide direction and information if associated value is limited or not. The limits are classified such as: Not Limited, High Limited, Low Limited, Constant.

When a input parameter is "connected" in a output parameter for cyclic data exchange, all structure (state and value) is received through fieldbus communication. If the input isnot connected, so states and values can be configured by user manually.

The state has following composition: Quality, sub-status and limits:

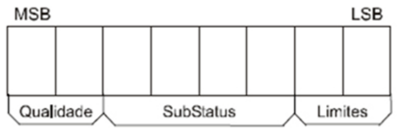


Figure 1 – Byte de Status

The state components are defined as below:

Quality - The quality used will be determined by the highest priority condition:

- 0 = Bad
- 1 = Uncertain
- 2 = Good (Non-Cascade)
- 3 = Good (Cascade)

Sub-status – The sub-status values in the atribute status are defined according to Table 1.

Limit – The following limits conditions Always will be available in the state.

- 0 = Not limited
- 1 = Low limit
- 2 = High limit
- 3 = Constants

Examples:

- 0xC1 (in hex) is the "Good-Cascade Non Specific and Low Limited" state
- 0xCF (in hex) is the "Good-Cascade Not invited and Constant" state
- 0x4E (in hex) is the "Uncertain Initial Value and High Limited" state.

Example: Number convertion for enumerations

There are many ways to convert the number listed to the state string. Two ways are show below:

1) Expressing the number in binary:

State of hex value = 78 = 0x4E = 01001110 (in binary)

Dividing this binary number in Quality, Sub-Status and Limit:

- Quality = 01 = 1 = "Uncertain"
- Sub-estado = 0011 = 3 = "Initial Value"
- Limit = 10 = 2 = "Limited in High"

The state is "uncertain - initial value - high limited."

2) Using the state value in decimal format:

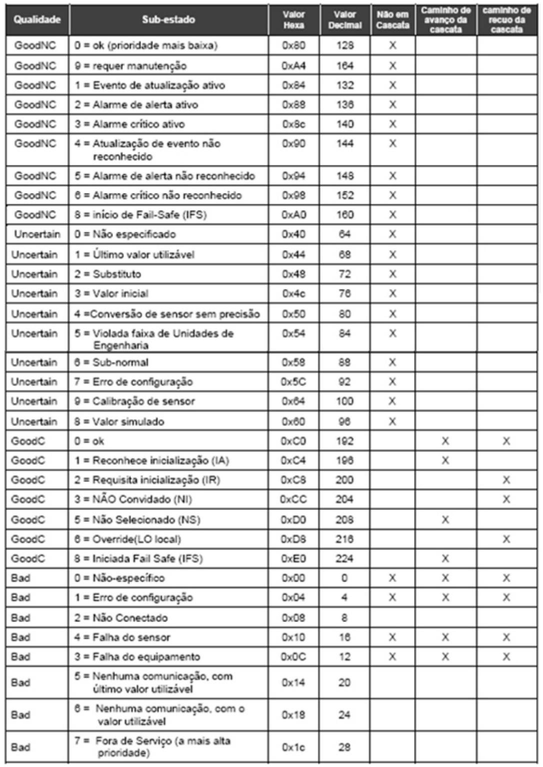
Decimal Value Status = 78

Divided the number per 64. The quotient will be Quality and stored the rest:

- Quality = 78 / 64 = 1
- Rest = 14

Divided the number per 4. The quotient will be Sub-Status and rest will be the limit:

- SubStatus = 14 / 4 = 3
- Limit = 2



Qualidade	Sub-estado	Valor Hexa	Valor Decimal	Não em Cascata	Caminho de avanço da cascata	caminho de recuo da cascata
GoodNC	0 = ok (prioridade mais baixa)	0x80	128	X		
GoodNC	9 = requer manutenção	0xA4	164	X		
GoodNC	1 = Evento de atualização ativo	0x84	132	X		
GoodNC	2 = Alarme de alerta ativo	0x88	136	X		
GoodNC	3 = Alarme crítico ativo	0x8c	140	X		
GoodNC	4 = Atualização de evento não reconhecido	0x90	144	X		
GoodNC	5 = Alarme de alerta não reconhecido	0x94	148	X		
GoodNC	6 = Alarme crítico não reconhecido	0x98	152	X		
GoodNC	8 = início de Fail-Safe (IFS)	0xA0	160	X		
Uncertain	0 = Não especificado	0x40	64	X		
Uncertain	1 = Último valor utilizável	0x44	68	X		
Uncertain	2 = Substituto	0x48	72	X		
Uncertain	3 = Valor inicial	0x4c	76	X		
Uncertain	4 =Conversão de sensor sem precisão	0x50	80	X		
Uncertain	5 = Violada faixa de Unidades de Engenharia	0x54	84	X		
Uncertain	6 = Sub-normal	0x58	88	X		
Uncertain	7 = Erro de configuração	0x5C	92	X		
Uncertain	9 = Calibração de sensor	0x64	100	X		
Uncertain	8 = Valor simulado	0x60	96	X		
GoodC	0 = ok	0xC0	192		X	X
GoodC	1 = Reconhece inicialização (IA)	0xC4	196		X	
GoodC	2 = Requisita inicialização (IR)	0xC8	200			X
GoodC	3 = NÃO Convidado (NI)	0xCC	204			X
GoodC	5 = Não Selecionado (NS)	0xD0	208		X	
GoodC	6 = Override(LO local)	0xD8	216			X
GoodC	8 = Iniciada Fail Safe (IFS)	0xE0	224		X	
Bad	0 = Não-específico	0x00	0	X	X	X
Bad	1 = Erro de configuração	0x04	4	X	X	X
Bad	2 = Não Conectado	0x08	8			
Bad	4 = Falha do sensor	0x10	16	X	X	X
Bad	3 = Falha do equipamento	0x0C	12	X	X	X
Bad	5 = Nenhuma comunicação, com último valor utilizável	0x14	20			
Bad	6 = Nenhuma comunicação, com o valor utilizável	0x18	24			
Bad	7 = Fora de Serviço (a mais alta prioridade)	0x1c	28			

Table 1 – Status Conditions

Manipulation in Fail Safe

> Fail Safe is a special state allow the function block to act when an abnormal situation is detected.
>
> An abnormal situation occurs when there is an unusual input, for example, faulty sensor or loss of communication between function blocks for a longer period than specified.
>
> When the condition that trigged the fault state is normalized, the Fault Status is reset and the block back to normal operation.

Condition that activate Fail Safe

> When the input and output Function Blocks detect an abnormal condition, the block enters in FAIL_SAFE mode. These abnormal situations are detected through different ways.
>
> The input function block, AI and TOT for example, are connected with a transducer block via channel. When the transducer output has a bad state (for example, faulty sensor) FAIL_SAFE condition of the block is activated.
>
> The output fucntion block, for example: AO block receive input values of the upper blocks through “cyclic connection”. These upper blocks are control function

> blocks and usually, have in control equipments. In this case, fail safe is activated when one of the following conditions is met:
>
> - Loss of communication RCAS_IN for a time that exceeds FSAFE_TIME;
> - Loss of communication SP during a time that exceeds FSAFE_TIME;
> - IFS state in RCAS_IN input, when RCas mode is designated;
> - IFS state in the SP, when Auto mode is designated.

Fail Safe Action

> The actions a imputo r output block can decide, when the block is in Fail Safe, should be selected by the user through FSAFE_TYPE parameter in the AI and AO blocks or using FAIL_TOT parameters in the TOT block.
>
> In the FSAFE_TYPE parameter, the following options are available:
>
> - Use FSAFE_VALUE – in this case, AI and AO blocks use the safety value supplied by the FSAFE_VALUE parameter for calculating, when Fail Safe is active. The output state change for "Uncertain, substitute value";
> - Use Last Usable Value – in this case, AI and AO blocks use the last usable value for algorithm calculation. The state will be "Uncertain Last Usable Level". Is there is still a satisfactory value, use na Initial Value in the output. The state will be "Uncertain Initial Value".
> - Use the wrong value" (only for AI block) – AI block use wrong values and states for calculation;
> - Use the ACTUATOR_ACTION (only for AO block) – AO block enter in safety position based on an ACTUATOR_ACTION discrete parameter in the transducer block.
>
> In the FAL_TOT parameter (only used in the totalizer block) the following options are available:
>
> - Hold – For totalizer in the last value. The state output will go to "Uncertain non-specific".
> - Memory – Use the last valid value for totalization. The state will be "Uncertain Last Usable Value". If there isnot a valid state in the memory, use the initial value for totalization. The state will be "Uncertain, Initial Value".
> - Run – The totalization is continued (restarted). Use the wrong value and state for output.

Conclusion

> We have seen through this article the importance of the status argument in Profibus PA and its peculiarities.

References:

1. SMAR Profibus Manuals
2. www.smar.com.br
3. Materiais de Treinamento Profibus – César Cassiolato