

# Siemens KTP-400

6AV2 123-2DB03-0AX0

## Alarm Configuration

- What are Alarms?
- What are the various Alarms?
- Setup Alarms
  - Discrete Alarms
  - Analogue Alarms
- Alarm Classes, Groups and Status



Code and Compile  
Learning Made Easy

[www.codeandcompile.com](http://www.codeandcompile.com)

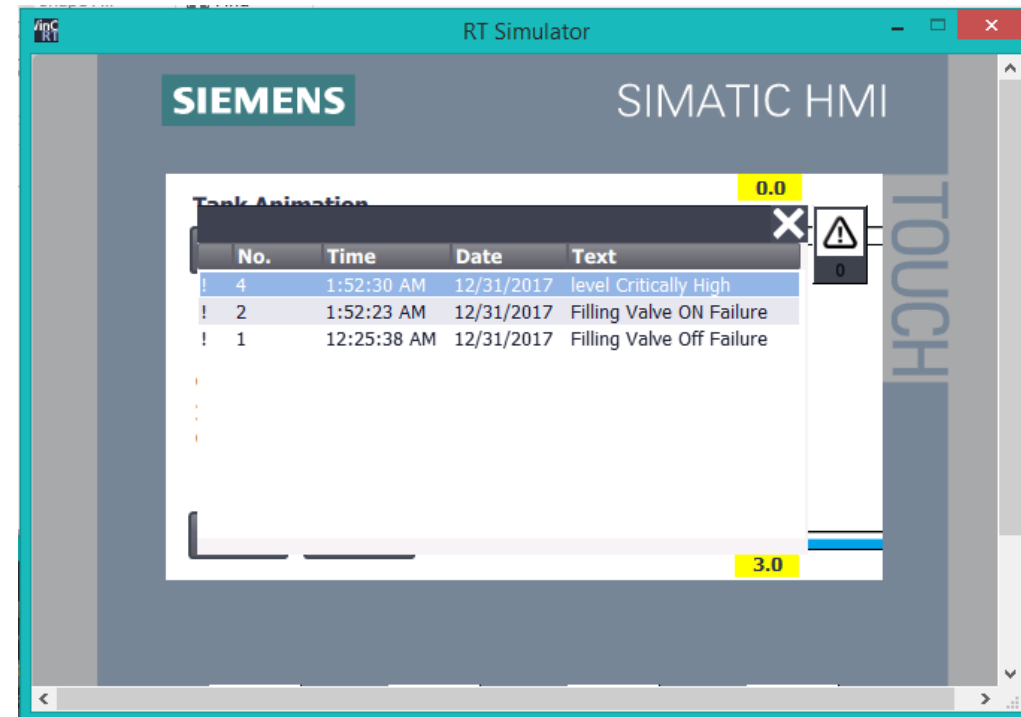
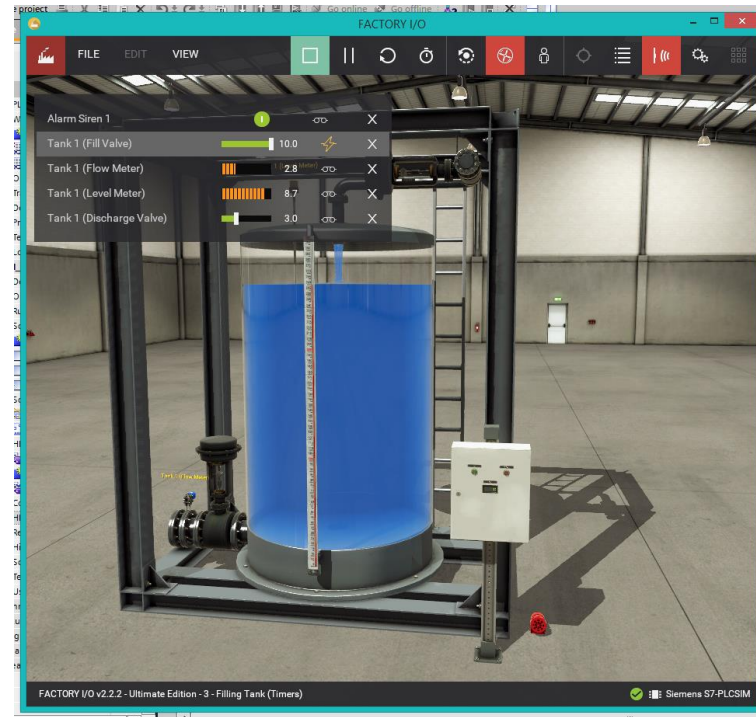




# What are Alarms?

audible, visual or other form of *alarm* signal about a problem or condition

**Example-** Critical level alarm in Liquid level control



## Alarm Devices:

- Sirens
- Visual Lights – Flash or LED's
- Visual Display – HMI or PC



# What are the various Alarms?

## Discrete Alarms

Digital alarms which are Trigger by a bit.

### Example:

- Safety limits
- Safety Sensors
- Sensor or Actuator
- Malfunctioning.

## Analogue Alarms

Analogue alarms which are Trigger by higher or lower Limit of any process variable.

### Example:

- Temperature limits
- Level limits

## System Alarms

These are system specific alarms For PLC and HMI.

### Example:

- PLC System faults
- Diagnostic alarms
- PLC-HMI Comm. error

## Monitoring Plant



## Monitoring Controllers





# Setup Digital Alarms in KTP400 HMI

Digital alarms are triggered by a bit inside a PLC. The most commonly used if your process does not involves analogue process variable.

Following are the steps involved:

1. Assign a word for Alarm(s) in PLC. This is required while setting up an alarm signals in HMI.
2. Write a logic in PLC and trigger a specific bit of that word for particular alarm signal.
3. For example, in Step 1 if you assign word MW5 for the alarm. In step 2 you need to assign M5.0, M5.1 .. and so on as alarm bit. You can use M5.0 for valve malfunctioning or M5.1 for motor malfunctioning etc.

## Example from PLC Tag table

	Alarm Word	Default tag table	Word	<u>%MW100</u> ✓
	Fillingonalarmbit	Default tag table	Bool	%M100.0 ✓
	fillingoffalarmbit	Default tag table	Bool	%M100.1 ✓

4. Open HMI Alarms > Discrete alarms in HMI tree and write specific message for the alarm with trigger address as MW100 as shown below

Discrete alarms							
ID	Alarm text	Alarm class	Trigger tag	Trigger bit	Trigger address	HMI acknowledgment tag	
1	Filling Valve Off Failure	Errors	Alarm Word	0	"Alarm Word".x0	<No tag>	
2	Filling Valve ON Failure	Errors	Alarm Word	1	"Alarm Word".x1	<No tag>	



# Setup Digital Alarms in KTP400 HMI



5. Make an independent Alarm Screen in HMI as shown below



The screenshot shows the SIMATIC Manager software interface. The main window displays a SIMATIC HMI screen with the following elements:

- Header: **SIEMENS** SIMATIC HMI
- Table with columns: No., Time, Date, Status, Text
- Button: Tank Animation
- Function keys: F1, F2, F3, F4

The right-hand side of the interface shows the **Toolbox** with the following sections:

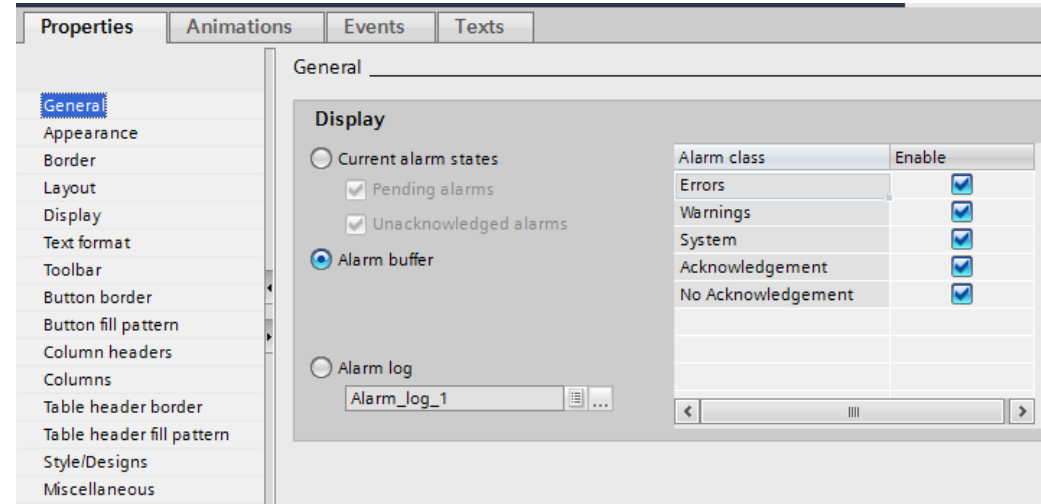
- Options:** Dark default value
- Basic objects:** Line, Ellip..., Circle, Rect..., Text..., Gra...
- Elements:** I/O..., Butt..., Sym..., Gra..., Dat..., Bar, Swit...
- Controls:** Alar... (highlighted with a red box), Tre..., Use..., HT..., Rec..., Syst...
- Graphics:** WnCC graphics folder, My graphics folder

A pink arrow points from the **Alar...** icon in the Controls section to the table area on the HMI screen.

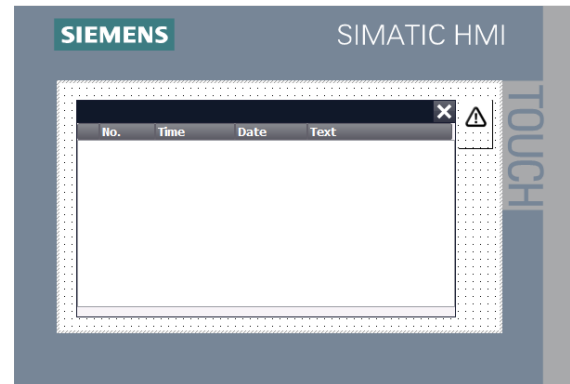
# Setup Digital Alarms in KTP400 HMI



6. Configure the alarm window in the properties as per requirements. More details in the video lecture



7. Add Alarm window in HMI Global screen for automatic pop up of alarms



*Navigate to Global screen via  
HMI > Screen Management > Global Screen*

8. That's it! Test your Alarms!

**Note:**  
You need to SWAP the byte for the Alarm Word  
IN PLC Programming to link it to HMI.

# Setup Analogue Alarms in KTP400 HMI



Analogue alarms are triggered when the process goes beyond the safety range. The safety range is defined by the user.

## Following are the steps involved:

1. In PLC, locate which is your process variable which you want to monitor for triggering alarms.
2. Open HMI Alarms > Analogue alarms in HMI tree and write specific message for the alarm with trigger tag as 'your process variable tag' as shown below:

ID	Alarm text	Alarm class	Trigger tag	Limit	Limit mode
1	Level Low	Warnings	tank level	2	Lower
2	Level Critically Low	Errors	tank level	1	Lower
3	Level High	Warnings	tank level	8	Higher
4	level Critically High	Errors	tank level	9	Higher
<Add new>					

3. Do the step 5,6 and 7 similar to Digital Alarm setup. That's it!



# Alarm Classes

Following are the various Alarm classes available.

Alarm classes								
	Display name	Name	State machine	Log	Backgro...	Backgro...	Backgro...	Backgro...
	!	Errors	Alarm with single-mode acknowledgment	<No log>		...		...
		Warnings	Alarm without acknowledgment	<No log>		255...		255...
	\$	System	Alarm without acknowledgment	<No log>		255...		255...
	A	Acknowledgement	Alarm with single-mode acknowledgment	<No log>		255...		255...
	NA	No Acknowledgement	Alarm without acknowledgment	<No log>		255...		255...

## It defines:

- How the Alarm should be displayed
- Specify If and how user has to acknowledge the alarm

## Examples:

1. Speed in tolerance range has 'Warning' class displayed in White and it does not have to be acknowledged
2. Speed exceeds upper warning range should have 'Error' class and should be pop up and displayed until user acknowledged that

# Alarm Groups

Various alarms can be assigned a common group to acknowledge group of alarms if required. It can have alarms from various classes. Ex: Alarm of same type – Tank Unit, Drive unit etc.







# Status of Alarms

Following are the various status of alarms:



S.No	Status	Description
1	I –Incoming	Alarm came ✓
2	IA- Incoming and Acknowledged ✓	<u>Alarm came</u> and <u>acknowledged</u> by user
3	IAO- <u>Incoming</u> , <u>Acknowledged</u> and <u>Outgoing</u>	Alarm came, acknowledged and cleared
4	<u>IO- Incoming</u> and <u>Outcoming</u> <u>without Acknowledged</u> ✗	Alarm came and cleared without acknowledged by user

# Thank you

*Get copy of this presentation  
in the course!*



**Code and Compile**  
Learning Made Easy

[www.codeandcompile.com](http://www.codeandcompile.com)

