

Backup Control

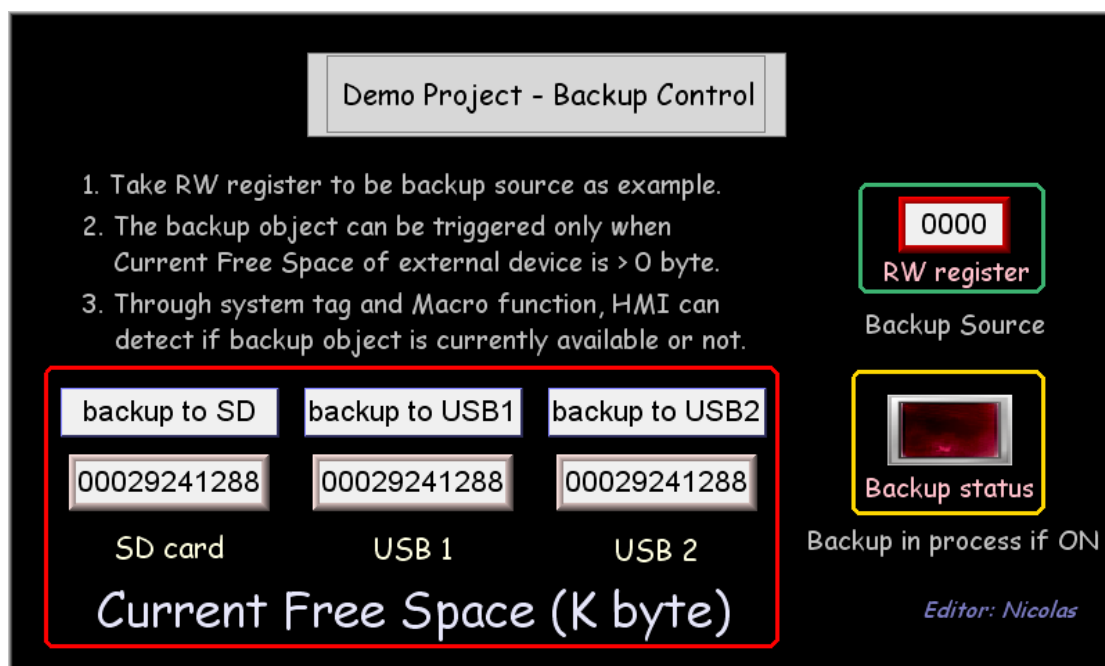
Table of Contents

1. Overview and Operation	3
2. Setting up the Screen	4
3. Addresses	8

1. Overview and Operation

Overview

Do you often feel unsure whether the Backup Object finishes backuping or not? Or accidentally trigger Backup Object before inserting External Device? System registers LW9074~LW9078 (External Devices current free space) allow the system to automatically determine if the Backup Object is currently triggerable.

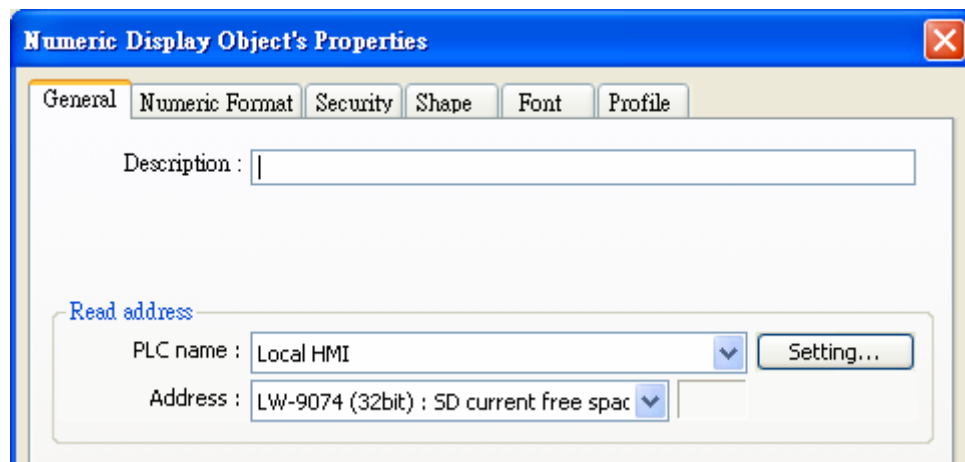


Operation

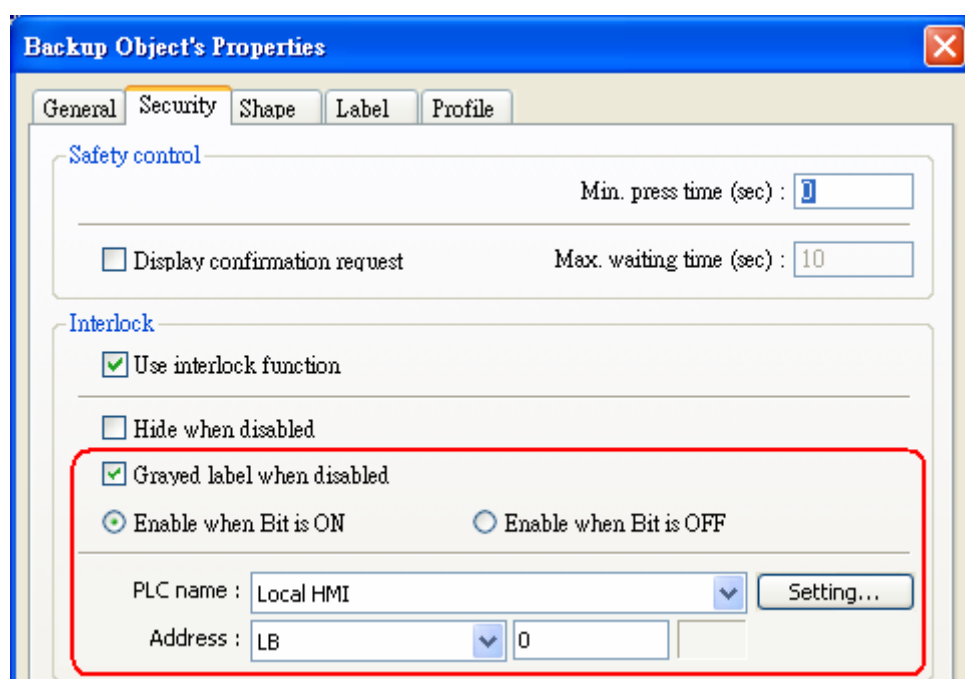
When inserting an External Device (Ex: SD card, USB1, and USB2) into HMI, via the preset relevant properties, that is MACRO interacting with interlock function of Backup Object, to automatically detect if any External Device is currently inserted . If an External Device is inserted and is with enough free space, Backup Object will be automatically enabled, otherwise, the function of triggering Backup Object will be disabled and will be grey labeled.

2. Setting up the Screen

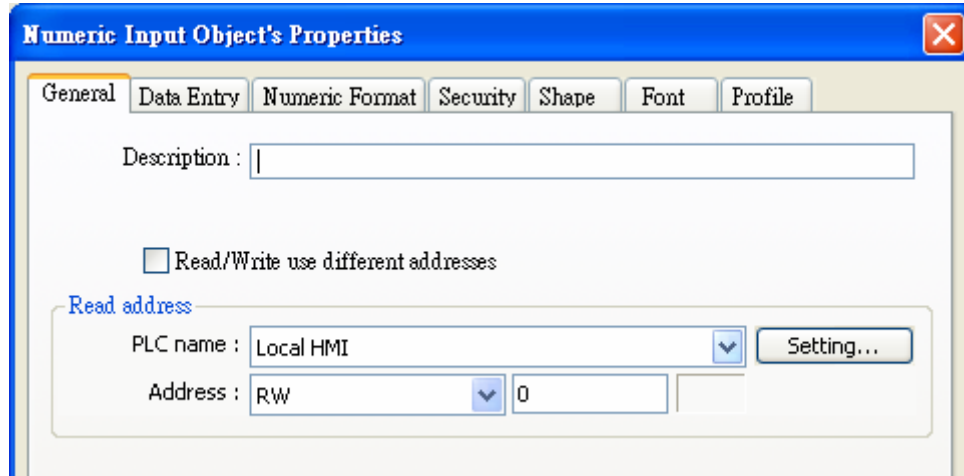
1. Firstly, 3 Numeric Display Objects must be created, with addresses LW9074, LW9076, and LW9078. They are displaying External Device current free space.



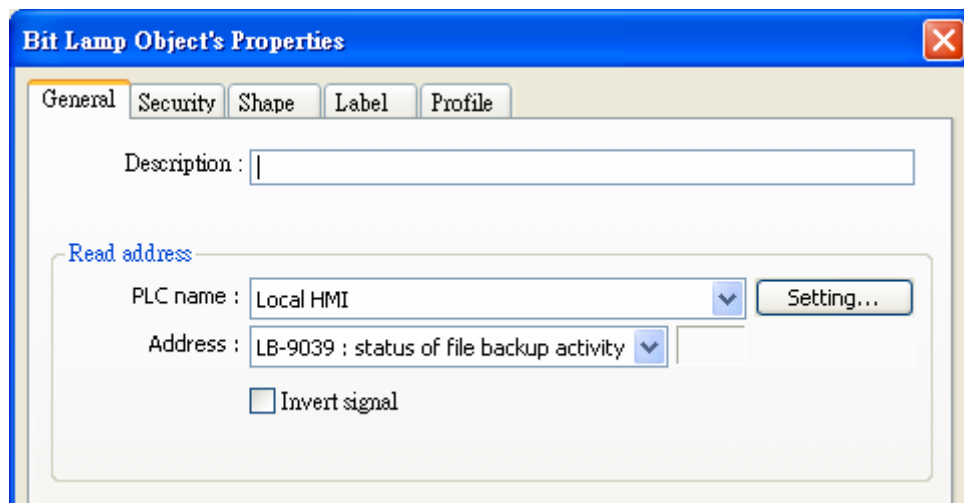
2. Create 3 Backup Objects, and set the target devices to SD card, USB1, and USB 2 respectively, and for each object set interlock function in [Security] tab. Please note that 3 Backup Objects respectively use 3 different Bit to enable / disable backup function.



3. You may create any backup source for backup function. In the following demonstration, a RW Numeric Input Object is built as backup source.



4. Create a Bit Lamp Object, with address LB9039, to be the indicator of backup activity status.



5. Edit MACRO to allow system to determine whether or not to trigger Backup Object.

Macro ID 1:

```
macro_command main()
```

```
int a, b, c
```

```
bool bit0=0, bit1=1 //status of interlock
```

```
GetData(a, "Local HMI", LW, 9074, 1) //SD memory
```

```
GetData(b, "Local HMI", LW, 9076, 1) //USB1 memory
```

```
GetData(c, "Local HMI", LW, 9078, 1) //USB2 memory
```

```
if a>0 then //SD memory
```

```
SetData(bit1, "Local HMI", LB, 0, 1) //enable backup
```

```
else
```

```
SetData(bit0, "Local HMI", LB, 0, 1) //disable backup
```

```
end if
```

```
if b>0 then //USB1 memory
```

```
SetData(bit1, "Local HMI", LB, 1, 1)
```

```
else
```

```
SetData(bit0, "Local HMI", LB, 1, 1)
```

```
end if
```

```
if c>0 then //USB2 memory
```

```
SetData(bit1, "Local HMI", LB, 2, 1)
```

```
else
```

```
SetData(bit0, "Local HMI", LB, 2, 1)
```

```
end if
```

```
end macro_command
```

6. Set a PLC Control Object to trigger MACRO.

PLC Control

Description :

PLC name : Local HMI

Attribute

Type of control : Execute macro program

☐ Active only when designated window opened

Macro name : macro_1 (ID : 1)

Trigger address

PLC name : Local HMI

Address : LB 10

Trigger mode : Always active when

- In the end, create a Set Bit Object to trigger MACRO always, and select "Set ON when window opens" for [Attribute].

Set Bit Object's Properties

General Security Shape Label Profile

Description :

Write address

PLC name : Local HMI

Address : LB 10

Attribute

Set style : Set ON when window opens

3. Addresses

The Object Addresses used in this demo project are listed below: Users can change Addresses and Object ID base on actual usage.

Object	Address	Object ID	Detail
Window 10			
Backup		BU_0	Backup to USB1
		BU_1	Backup to USB2
		BU_2	Backup to SD card
Numeric Input	RW0	NE_0	Backup source
Numeric Display	LW9074	ND_0	SD card memory
	LW9076	ND_1	USB1 memory
	LW9078	ND_2	USB2 memory
Bit Lamp	LB9039	BL_0	Backup status
Set Bit	LB10	SB_0	Trigger MACRO
PLC Control	LB10		Execute MACRO