How to use the custom font

In this tutorial, we will show you how to use the custom font on the Y Series module. Y Series module will retain the information in the RAM until initialized command, reset command is sent or when the power is off.

Part 1 - Convert the design into data ►

Part 2 - Store the custom font ►

Part 3 - Enable / disable the custom fonts ►

Conversion templates ►

How to convert the custom font

In this tutorial, we will show you how to convert custom font into data for the Y Series module. You will first need to create a 5x8 matrix design then convert the design into hexadecimal data using the conversion table.

Step 1 - Create a 5x8 design

Design a custom font in 5x8 matrixes. In this tutorial we will be converting the following design.

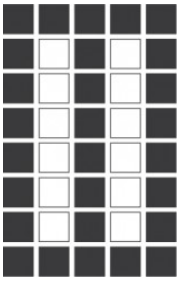
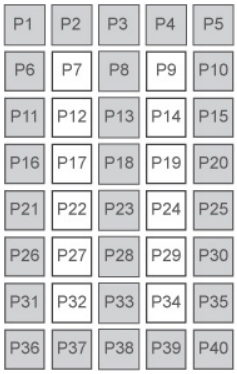


Fig. 1 Sample 5x8 custom font

Step 2 - Convert the design into data

Convert the design using the conversion table below.



Conversion Table	Bit							
	7	6	5	4	3	2	1	0
Byte 1	P8	P7	P6	P5	P4	P3	P2	P1
Byte 2	P16	P15	P14	P13	P12	P11	P10	P9
Byte 3	P24	P23	P22	P21	P20	P19	P18	P17
Byte 4	P32	P31	P30	P29	P28	P27	P26	P25
Byte 5	P40	P39	P38	P37	P36	P35	P34	P33

Converted	Bit							
	7	6	5	4	3	2	1	0
Byte 1	1	0	1	1	1	1	1	1
Byte 2	1	1	0	1	0	1	1	0
Byte 3	0	1	0	1	1	0	1	0
Byte 4	0	1	1	0	1	0	1	1
Byte 5	1	1	1	1	1	1	0	1

Fig. 2 Conversion table with sample conversion

You should get the following values with the design used in this example:
Byte 1: **BFh**
Byte 2: **D6h**
Byte 3: **5Ah**
Byte 4: **6Bh**
Byte 5: **FDh**

How to store the custom font

In this tutorial, we will show you how to store the custom font into **Y Series modules** random-access memory (RAM). Y Series can store up to 16 custom fonts between character codes 20h and FFh.

Step 1 - Store your design into RAM

In this step, we will be using the design created and converted from "**How to convert custom font**":

P1	P2	P3	P4	P5
P6	P7	P8	P9	P10
P11	P12	P13	P14	P15
P16	P17	P18	P19	P20
P21	P22	P23	P24	P25
P26	P27	P28	P29	P30
P31	P32	P33	P34	P35
P36	P37	P38	P39	P40

Conversion Table	Bit							
	7	6	5	4	3	2	1	0
Byte 1	P8	P7	P6	P5	P4	P3	P2	P1
Byte 2	P16	P15	P14	P13	P12	P11	P10	P9
Byte 3	P24	P23	P22	P21	P20	P19	P18	P17
Byte 4	P32	P31	P30	P29	P28	P27	P26	P25
Byte 5	P40	P39	P38	P37	P36	P35	P34	P33

Converted	Bit							
	7	6	5	4	3	2	1	0
Byte 1	1	0	1	1	1	1	1	1
Byte 2	1	1	0	1	0	1	1	0
Byte 3	0	1	0	1	1	0	1	0
Byte 4	0	1	1	0	1	0	1	1
Byte 5	1	1	1	1	1	1	0	1

Fig. 1 Conversion table with sample conversion

Converted data:
Byte 1: **BFh**
Byte 2: **D6h**
Byte 3: **5Ah**
Byte 4: **6Bh**
Byte 5: **FDh**

Send the following codes to store the design above into character code 21h:

1	1Bh 26h 01h 21h 21h 05h BFh D6h 5Ah 6Bh FDh

Codes breakdown

Line #1	1Bh 26h 01h 1	21h 21h 2	05h 3	BFh D6h 5Ah 6Bh FDh 4

1. Stores the custom font into the Y Series module's RAM.
   Sets the start and end character code locations.
   When storing a custom font, the start and end value is the same.
2. Fixed, sets the width of the custom character
3. The converted data in order of Byte 1 to Byte

How to enable / disable the custom fonts

In this tutorial, we will show you how to enable/disable custom fonts stored in the RAM of the **Y Series module**. The custom fonts are disabled by default; therefore the custom fonts need to be enabled first. To display default font, please disable the custom fonts.

Step 1 - Enable the custom fonts

Send the following command to enable the custom fonts:

```
1Bh 25h 01h
```

Step 2 - Disable the custom fonts

Send the following command to disable the custom fonts:

```
1Bh 25h 00h
```

This tutorial is based on "Initial setting", "RAM user font Define" and "RAM user font" in the specification.
Download the Y Series spec ►

Conversion templates

P1	P2	P3	P4	P5
P6	P7	P8	P9	P10
P11	P12	P13	P14	P15
P16	P17	P18	P19	P20
P21	P22	P23	P24	P25
P26	P27	P28	P29	P30
P31	P32	P33	P34	P35
P36	P37	P38	P39	P40

Conversion Table	Bit							
	7	6	5	4	3	2	1	0
Byte 1	P8	P7	P6	P5	P4	P3	P2	P1
Byte 2	P16	P15	P14	P13	P12	P11	P10	P9
Byte 3	P24	P23	P22	P21	P20	P19	P18	P17
Byte 4	P32	P31	P30	P29	P28	P27	P26	P25
Byte 5	P40	P39	P38	P37	P36	P35	P34	P33

Byte 1:
Byte 2:
Byte 3:
Byte 4:
Byte 5:

P1	P2	P3	P4	P5
P6	P7	P8	P9	P10
P11	P12	P13	P14	P15
P16	P17	P18	P19	P20
P21	P22	P23	P24	P25
P26	P27	P28	P29	P30
P31	P32	P33	P34	P35
P36	P37	P38	P39	P40

Conversion Table	Bit							
	7	6	5	4	3	2	1	0
Byte 1	P8	P7	P6	P5	P4	P3	P2	P1
Byte 2	P16	P15	P14	P13	P12	P11	P10	P9
Byte 3	P24	P23	P22	P21	P20	P19	P18	P17
Byte 4	P32	P31	P30	P29	P28	P27	P26	P25
Byte 5	P40	P39	P38	P37	P36	P35	P34	P33

Byte 1:
Byte 2:
Byte 3:
Byte 4:
Byte 5:

P1	P2	P3	P4	P5
P6	P7	P8	P9	P10
P11	P12	P13	P14	P15
P16	P17	P18	P19	P20
P21	P22	P23	P24	P25
P26	P27	P28	P29	P30
P31	P32	P33	P34	P35
P36	P37	P38	P39	P40

Conversion Table	Bit							
	7	6	5	4	3	2	1	0
Byte 1	P8	P7	P6	P5	P4	P3	P2	P1
Byte 2	P16	P15	P14	P13	P12	P11	P10	P9
Byte 3	P24	P23	P22	P21	P20	P19	P18	P17
Byte 4	P32	P31	P30	P29	P28	P27	P26	P25
Byte 5	P40	P39	P38	P37	P36	P35	P34	P33

Byte 1:
Byte 2:
Byte 3:
Byte 4:
Byte 5: