

vlw Font Creator

vlw Font Creator is an online vlw font generator, used to generate customized pixel size and font range of vlw font files, easy for developers to expand the display font style and save the font memory consumption.

1.Create vlw fonts

Visit <https://vlw-font-creator.m5stack.com/>, follow the instructions below to complete vlw font creation.

The screenshot shows the vlw Font Creator web interface. It has a light blue border and a white background. The form includes the following fields and instructions:

- Names ***: A text input field containing "Test". A red instruction "① Font name; Same within the code." is shown.
- Size ***: A text input field containing "20". A red instruction "② Font size in pixels." is shown.
- ☐ **Try to use glyph color info from font to create grayscale icons.**
Since gray tones are emulated via transparency, result will be good on contrast background only.
- TTF/WOFF/OTF File ***: A blue button "Choose file" and a text input field containing "Ubuntu-C.ttf". A red instruction "③ Select your font file." is shown.
- Preview Text**: A text input field containing "input text or unicode(\uXXXX) to preview".
- Preview**: A text input field containing "quick fox jumps over the lazy dog" and "You can preview your font here.".
- Range ***: A text input field containing "0x20-0x7f". A red instruction "④ Unicode block range , optional." is shown.
- Symbols ***: A text input field containing "HelloWorld! 你好世界!". A red instruction "⑤ Specify the character, optional." is shown.
- A red instruction "* At least one of ④ and ⑤ must be specified."
- A blue button "Include another font".
- A text input field containing "You can use both "Range" and "Symbols" or only one of them".
- A blue button "Convert".
- A red instruction "⑥ Get your vlw font."

2.Using vlw fonts

Refer to the following case study, and reference the vlw font file to your project.

1. Using vlw font file stored in chip memory

- o [Arduino - M5GFX vlw Font](#)

2. Using vlw font file stored in microSD

```
1  #include <Arduino.h>
2  #include <SPI.h>
3  #include <SD.h>
4  #include <M5Unified.h>
5  #include <M5GFX.h>
6
7  #define SD_SPI_CS_PIN 4
8  #define SD_SPI_SCK_PIN 18
9  #define SD_SPI_MISO_PIN 19
10 #define SD_SPI_MOSI_PIN 23
11
12 void setup() {
13     M5.begin();
14
15     M5.Display.setTextFont(&fonts::Orbitron_Light_24);
16     M5.Display.setTextSize(1);
17
18     // SD Card Initialization
19     SPI.begin(SD_SPI_SCK_PIN, SD_SPI_MISO_PIN, SD_SPI_MOSI_PIN, SD_SPI_CS_PIN);
20     if (!SD.begin(SD_SPI_CS_PIN, SPI, 25000000)) {
21         // Print a message if SD card initialization failed or if the SD card does not exist.
22         M5.Display.print("\n SD card not detected\n");
23         while (1)
24             ;
25     } else {
26         M5.Display.print("\n SD card detected\n");
27     }
28     delay(1000);
29
30     M5.Display.print("\n SD card read test...\n");
31     if (SD.open("/Test.vlw", FILE_READ, false)) {
32         M5.Display.print(" vlw detected\n");
33     } else {
34         M5.Display.print(" vlw not detected\n");
35     }
36
37     if (M5.Display.loadFont(SD, "/Test.vlw")) {
38         M5.Display.print("\n↕↔@"); // Display some characters to test the font
39     } else {
40         M5.Display.print(" vlw not loaded\n");
41     }
42 }
43
44 void loop() {
45 }
```

The example program runs as follows:

