

list13.txt

入力: 上記プログラムで、フォントの大きさを縦横とも2倍にしてください。

GPT:

フォントの大きさを縦横ともに2倍にするためには、各ピクセルを2x2のブロックで表示します。以下の手順で進めていきます。

1. 2倍の大きさを表示する関数の追加

まず、'oled_putc'関数を変更して、フォントを2倍に拡大して表示する新しい関数'oled_putc_2x'を作成します。

<リスト7>

```
void oled_putc_2x(char c) {
    if (c < 0x20 || c > 0x7F) return;

    uint8_t buffer[11]; // 5x2 + 1 (OLED_DATA)
    buffer[0] = OLED_DATA; // Data mode

    for (int i = 0; i < 5; i++) {
        uint8_t ch_data = Font5x8[c - 0x20][i];
        uint8_t ch_data_expanded1 = 0;
        uint8_t ch_data_expanded2 = 0;

        for (int j = 0; j < 8; j++) {
            if (ch_data & (1 << j)) {
                if (j < 4) {
                    ch_data_expanded1 |= (3 << (j*2)); // 2倍拡大の下半分
                } else {
                    ch_data_expanded2 |= (3 << ((j-4)*2)); // 2倍拡大の上半分
                }
            }
        }

        buffer[i*2 + 1] = ch_data_expanded1;
        buffer[i*2 + 2] = ch_data_expanded2;
    }

    i2c_write_blocking(i2c0, OLED_ADDR, buffer, 11, false);

    // 同じデータを1行下にも表示して、縦方向に2倍の高さにする
    oled_set_pos((buffer[1] >> 3) + 1, (buffer[0] & 0x07));
    i2c_write_blocking(i2c0, OLED_ADDR, buffer, 11, false);
}
```

1. 文字列表示関数の変更

既存の'oled_puts'関数を使用して、新しく作成した'oled_putc_2x'関数を使用するように変更します。

```
void oled_puts_2x(const char* str) {
    while (*str) {
        oled_putc_2x(*str++);
    }
}
```

1. 'main'関数での呼び出しの変更

最後に、'main'関数内での文字列表示関数の呼び出しを'oled_puts'から'oled_puts_2x'に変更します。

```
int main() {
    stdio_init_all();
}
```

list13.txt

```
oled_init();
oled_clear(); // 画面内容を消去

oled_set_pos(0, 0); // Set cursor to top-left corner
oled_puts_2x("Hello, OLED!");

while (true) {
    tight_loop_contents();
}

return 0;
}
```

以上の変更を行うことで、文字を縦横2倍の大きさで表示することができます。