

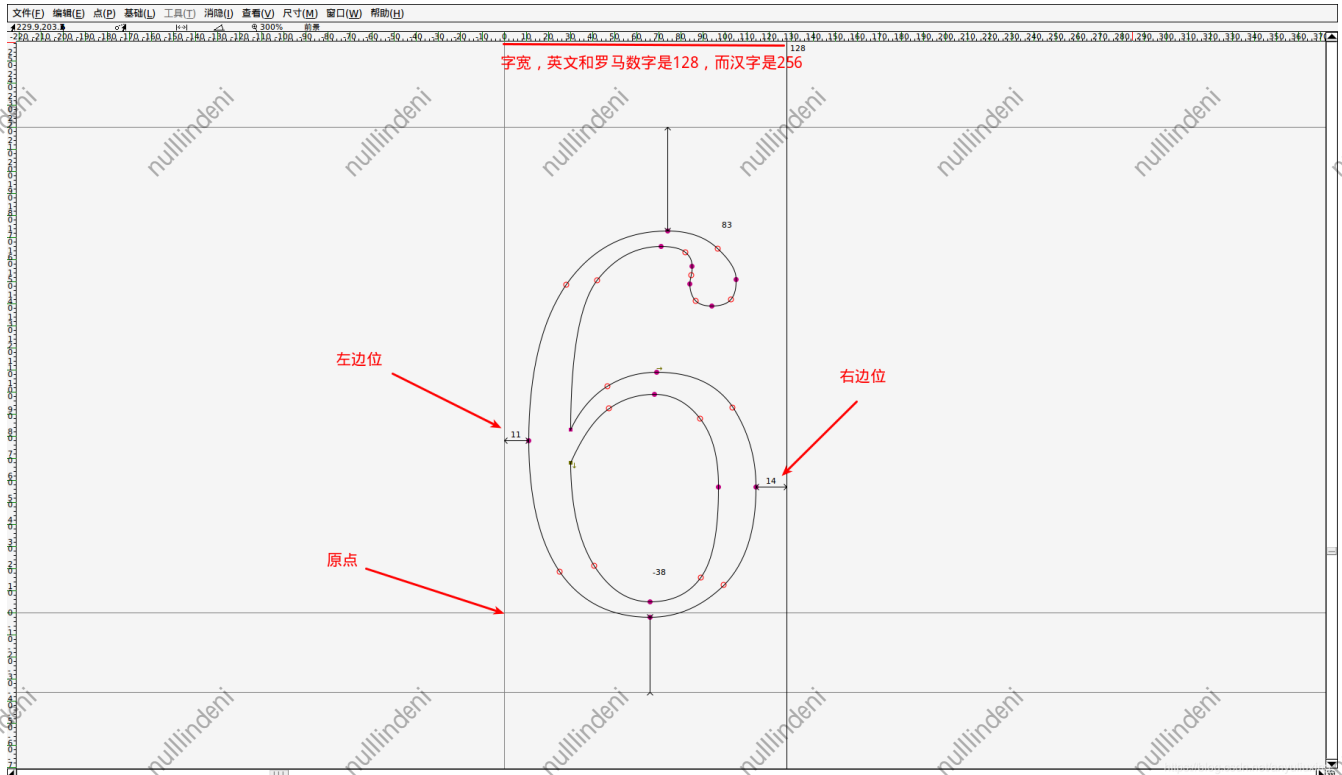
3. 点击上面工具栏的基础-->变换-->如图设置



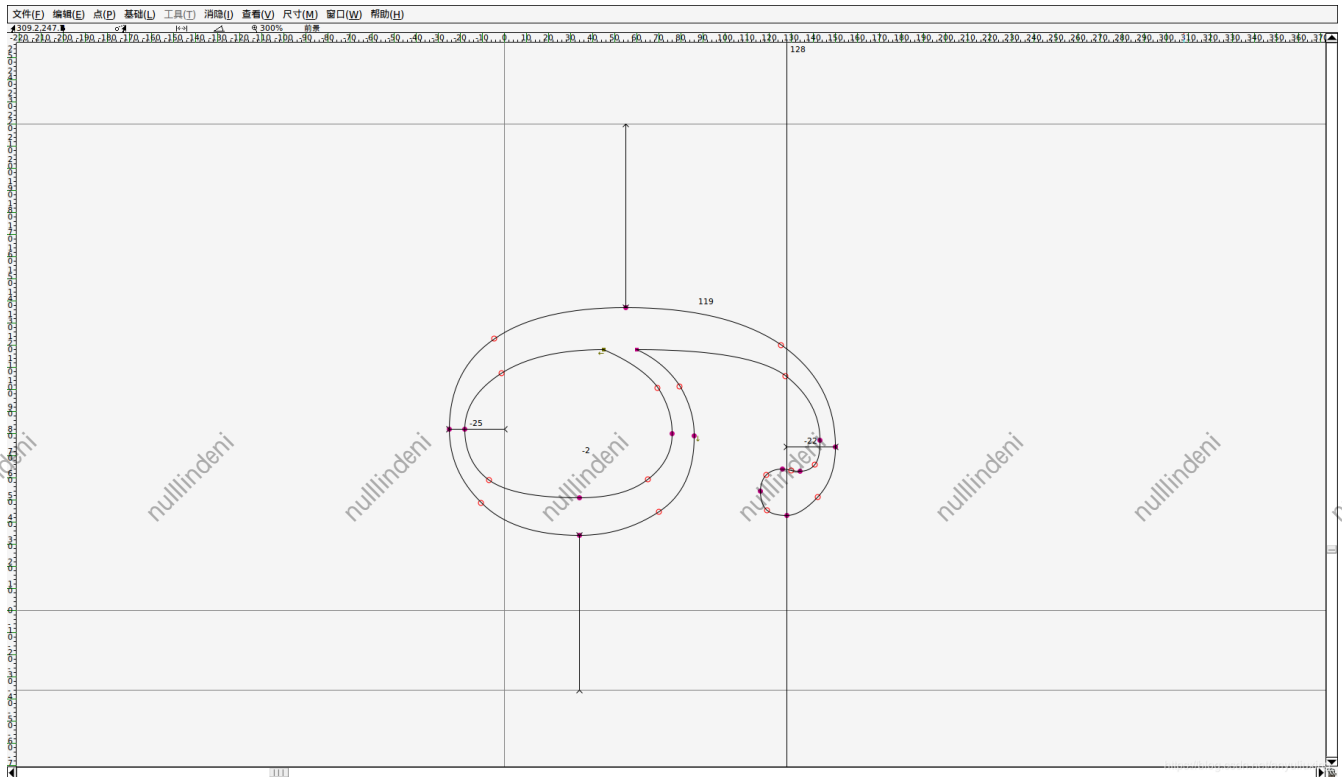
需要注意的是

1. 需要选择字形原点，不能按选中部分的中心旋转，不然会导致汉字与数字和英文显示的高度不在同一水平线上，如果按选中部分的中心旋转，第3步换成第4步
2. 需要选择顺时针旋转90度，如果逆时针旋转90度的话，文字显示会由“你好”变成“好你”，并且对齐方式是向右对齐，需要做一些额外的处理
3. 需要选择向Y轴移动220个单位，不同字体可能需要移动的数字不同，可以在MiniGUI中用TextOut函数设置y的值为0，看文字是否显示在y为0的位置上，动态调试
4. 第2步旋转之后，只选择数字和英文，点击基础-->变换，选择字形原点，选择向Y轴移动100个单位，之后点击尺寸-->设置左边位，设置好数值，然后在MiniGUI中观看效果，动态调试
5. 因为MiniGUI的TextOut，DrawText函数不支持竖直输出文本，需要修改MiniGUI源码，使文字竖直输出

这里对上面的一些概念做一些说明，还没有旋转之前

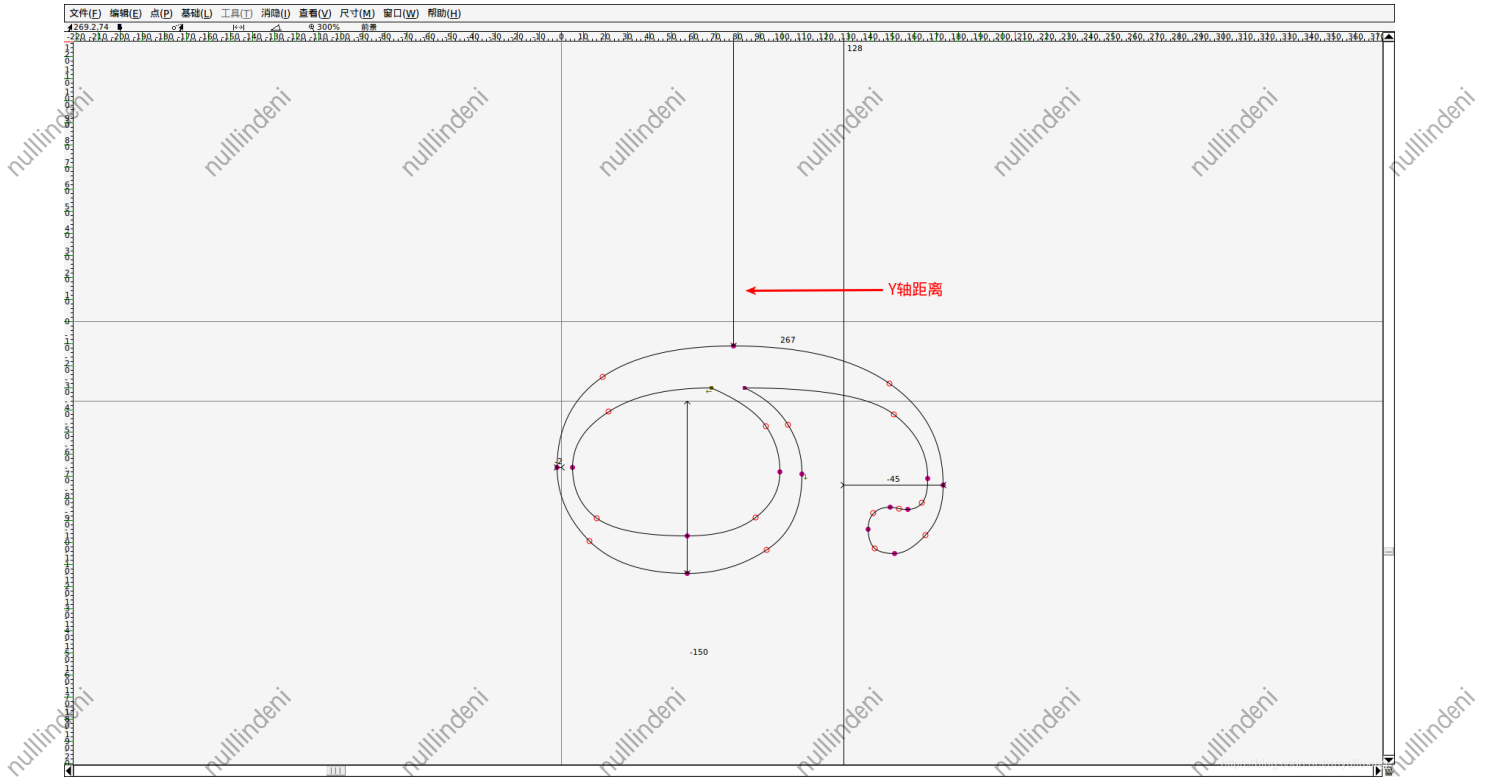


当按中心旋转之后，左边位变成-25，所以需要调整左边位，不然数字与汉字显示不在同一水平线上，因为汉字宽高都是256，所以汉字不需要调整左边位

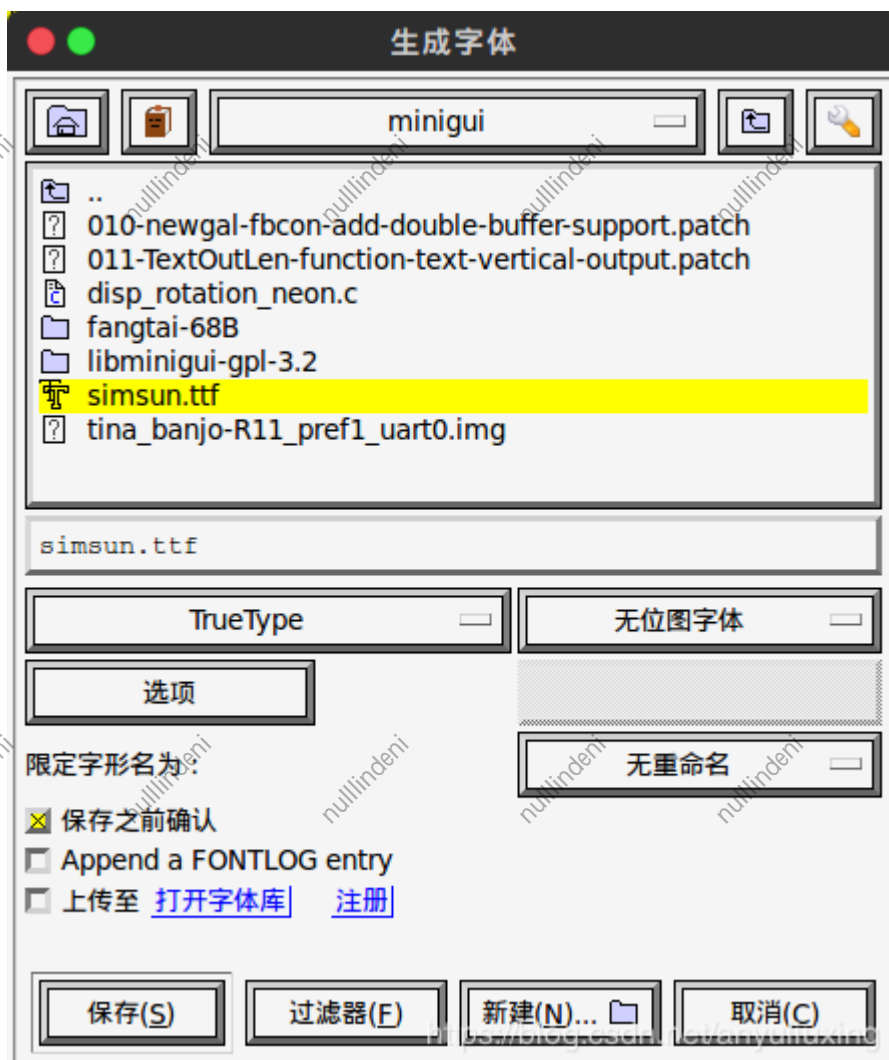


当按原点旋转之后，Y轴距离变大了很多，所以需要向上移动一定的距离，不然文字显示的时候会空出一些，因为是按原点旋转，汉字与数字可以显示在同一水平线上，因此可以不用改动左边位，但是我们可以看到，左边位也是变

小了，也可以根据需要修改



4.文件-->生成字体-->TrueType



二. MiniGUI 垂直显示文本

主要是把下一次输出的x, y轴的坐标对换就可以了, 新增Align

宏	描述
TA_HORIZONTAL	水平输出
TA_VERTICAL_CW	垂直输出, 字库顺时针旋转
TA_VERTICAL_CCW	垂直输出, 字库逆时针旋转

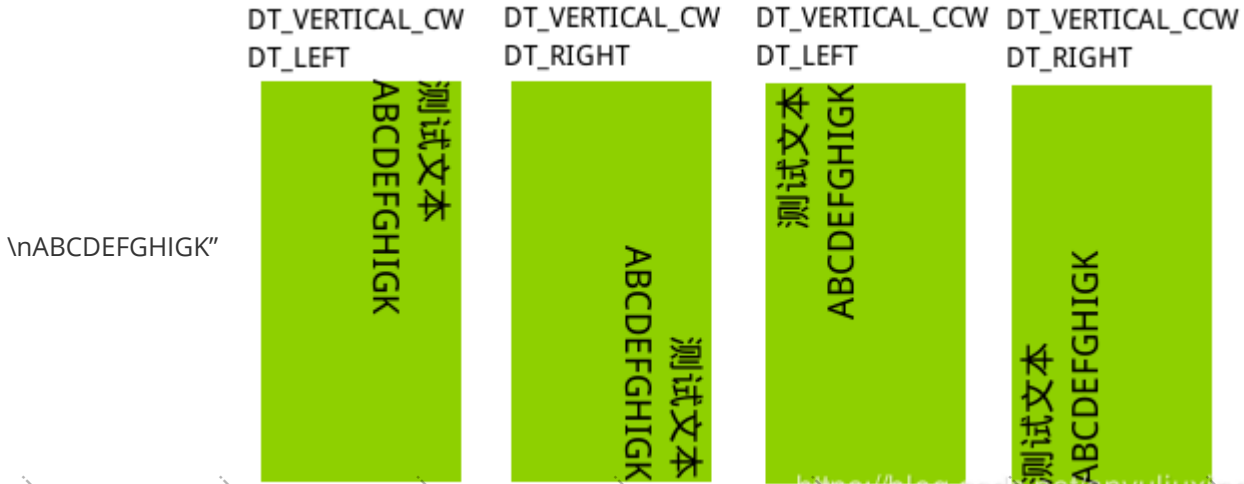
SetTextAlign只影响当前hdc, 对其他界面无影响 在调用TextOut和TabbedTextOut函数之前调用 SetTextAlign(hdc,TA_VERTICAL_CW), 调用SetTextAlign(hdc,TA_HORIZONTAL)可以恢复该hdc为水平输出文字

DrawText不受SetTextAlign的影响, 新增Align

宏	描述
DT_VERTICAL_CW	垂直输出, 字库顺时针旋转
DT_VERTICAL_CCW	垂直输出, 字库逆时针旋转

在调用DrawText的时候可以指定文字输出风格，不指定这两个风格为水平输出

为什么需要区分字库顺时针旋转和逆时针旋转，这里主要涉及到换行的问题，水平输出的情况下，换行是Y轴加一个字的高度。竖直输出，字库顺时针旋转的情况下是X轴减少一个字的高度。竖直输出，字库逆时针旋转的情况下是X轴增加一个字的高度，并且需要注意，想要输出“你好啊”，源字符串需要是“啊你好”，如果是两行的话，还需要向右对齐，当然MiniGUI中是支持双向文本的，补丁中已经做好了支持，具体效果如下，以下的字符串都是“测试文本”



TextOut和TabbedTextOut指定了TA_VERTICAL_CCW之后，字符串也是不用手动逆转 原本DrawText指定DT_RIGHT和竖直输出文本之后，位置和设定的位置会有偏差，补丁中也已经修复 需要注意的是DrawText指定竖直输出后，DT_BOTTOM，DT_VCENTER，DT_CENTER不能再使用

使用示例如下

```
case MSG_PAINT: {
    HDC hdc = BeginPaint(hWnd);

    SetTextAlign(hdc, TA_VERTICAL_CW);
    TextOut(hdc, 10, 150, "你好啊");
    TabbedTextOut(hdc, 200, 300, "你好啊");

    RECT rcClient;
    rcClient.left = 10;
    rcClient.top = 240;
    rcClient.right = 400;
    rcClient.bottom = 400;
    DrawText(hdc, "你好啊", -1, &rcClient,
             DT_NOCLIP | DT_LEFT | DT_WORDBREAK | DT_VERTICAL_CW);

    TextOut(hdc, 250, 140, "2:45");
    EndPaint(hWnd, hdc);
    return 0;
}
```

关键点说明

DrawText函数，如果设置了竖直输出，则每绘制一个字，对调X，Y的坐标

```

diff --git a/src/newgdi/drawtext.c b/src/newgdi/drawtext.c
index 6f003b8..42e2290 100644
--- a/src/newgdi/drawtext.c
+++ b/src/newgdi/drawtext.c
@@ -370,8 +370,14 @@ static BOOL cb_drawtextex2 (void* context, Glyph32 glyph_value,
     break;
 }

-   ctxt->x += adv_x;
-   ctxt->y += adv_y;
+   if (!(ctxt->nFormat & DT_VERTICAL_CW)
+       && !(ctxt->nFormat & DT_VERTICAL_CCW)) {
+       ctxt->x += adv_x;
+       ctxt->y += adv_y;
+   } else {
+       ctxt->x += adv_y;
+       ctxt->y += adv_x;
+   }

    return TRUE;
}

```

DrawText函数，如果设置了竖直输出，则对输出区域的起点X，Y进行调整，使其位置不偏离

```

@@ -476,14 +483,25 @@ int DrawTextEx2 (HDC hdc, const char* pText, int nCount,
    }

    /* set the start_x pos.*/
-   if (nFormat & DT_RIGHT) {
-       x = rcDraw.right;
-       old_ta = SetTextAlign(hdc, TA_RIGHT);
-   }
-   else {
-       old_ta = SetTextAlign(hdc, TA_LEFT);
-       x = rcDraw.left;
-   }
+   if (nFormat & DT_RIGHT) {
+       x = rcDraw.right;
+       old_ta = SetTextAlign(hdc, TA_RIGHT);
+       if (nFormat & DT_VERTICAL_CW) {
+           x = rcDraw.right + (pdc->pLogFont->size >> 1);
+           y = rcDraw.bottom - pdc->pLogFont->size;
+       } else if (nFormat & DT_VERTICAL_CCW) {
+           x = rcDraw.left + pdc->pLogFont->size + (pdc->pLogFont->size >> 1);
+           y = rcDraw.bottom - pdc->pLogFont->size;
+       }
+   } else {
+       old_ta = SetTextAlign(hdc, TA_LEFT);
+       x = rcDraw.left;
+       if (nFormat & DT_VERTICAL_CW) {
+           x = rcDraw.right - (pdc->pLogFont->size >> 1);
+       } else if (nFormat & DT_VERTICAL_CCW) {
+           x = rcDraw.left + pdc->pLogFont->size - (pdc->pLogFont->size >> 1);

```

```

+     }
+ }

    if(nFormat & DT_CALCRECT){
        *pRect = rcDraw;
    }
}

```

DrawText函数，如果设置了竖直输出，转换输出方向，使应用不用对字符串进行逆转

```

@@ -503,6 +521,11 @@ int DrawTextEx2 (HDC hdc, const char* pText, int nCount,
    ctxt.x = x;
    ctxt.y = y;

+   if (nFormat & DT_VERTICAL_CCW)
+       direction = (pdc->ta_flags & TA_X_MASK) == TA_RIGHT;
+   else
+       direction = (pdc->ta_flags & TA_X_MASK) != TA_RIGHT;
+
    while (nCount > 0) {
        int line_x, maxwidth;
    }
}

```

DrawText函数，指定了DT_VERTICAL_CW向左换行，DT_VERTICAL_CCW向右换行，水平输出是向下换行

```

@@ -597,13 +617,25 @@ int DrawTextEx2 (HDC hdc, const char* pText, int nCount,
    line_len -= ctxt.nCount;
    pline += ctxt.nCount;
-   y += ctxt.line_height;
+   if (nFormat & DT_VERTICAL_CW) {
+       ctxt.start_x -= ctxt.line_height;
+   } else if (nFormat & DT_VERTICAL_CCW) {
+       ctxt.start_x += ctxt.line_height;
+   } else {
+       y += ctxt.line_height;
+   }
    nLines ++;
}

/* continuous multiline '\n'.*/
if ((nr_delim_newline-1) > 0){
-   y += ctxt.line_height * (nr_delim_newline-1);
+   if (nFormat & DT_VERTICAL_CW) {
+       ctxt.start_x -= ctxt.line_height * (nr_delim_newline - 1);
+   } else if (nFormat & DT_VERTICAL_CCW) {
+       ctxt.start_x += ctxt.line_height * (nr_delim_newline - 1);
+   } else {
+       y += ctxt.line_height * (nr_delim_newline - 1);
+   }
    nLines += (nr_delim_newline-1);
}
}

```

TextOut和TabbedTextOut也是类似的处理，不再赘述

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni

nullindeni