

```

//固件修改更新QQ群: 224981313
//固件配套的3D打印机学习套件购买地址: http://item.taobao.com/item.htm?id=42916612391
//网盘资料: http://yunpan.taobao.com/s/19pI3jQSt0L
//视频教程 - v1,v2版机型软件设置, 自动调平
//优酷地址: http://www.youku.com/playlist\_show/id\_23218776.html
//视频教程 - v3版机型组装部分, 即初八以后发货的组装视频
//优酷地址: http://www.youku.com/playlist\_show/id\_23522533.html

#ifndef CONFIGURATION_H
#define CONFIGURATION_H

//=====
//=====MICROMAKE 3D打印机配套固件 =====
//=====
#define STRING_VERSION_CONFIG_H __DATE__ " " __TIME__
#define STRING_CONFIG_H_AUTHOR "(MICROMAKE KOSSEL)"

#define SERIAL_PORT 0

// 串口通讯速率
#define BAUDRATE 115200

//// 驱动板类型
// 10 = Gen7 custom (Alfons3 Version) "https://github.com/Alfons3/Generation\_7\_Electronics"
// 11 = Gen7 v1.1, v1.2 = 11
// 12 = Gen7 v1.3
// 13 = Gen7 v1.4
// 2 = Cheaptronic v1.0
// 20 = Sethi 3D_1
// 3 = MEGA/RAMPS up to 1.2 = 3
// 33 = RAMPS 1.3 / 1.4 (Power outputs: Extruder, Fan, Bed)
// 34 = RAMPS 1.3 / 1.4 (Power outputs: Extruder0, Extruder1, Bed)
// 35 = RAMPS 1.3 / 1.4 (Power outputs: Extruder, Fan, Fan)
// 4 = Duemilanove w/ ATmega328P pin assignment
// 5 = Gen6
// 51 = Gen6 deluxe
// 6 = Sanguinololu < 1.2
// 62 = Sanguinololu 1.2 and above
// 63 = Melzi
// 64 = STB V1.1
// 65 = Azteeg X1
// 66 = Melzi with ATmega1284 (MaKr3d version)
// 67 = Azteeg X3
// 68 = Azteeg X3 Pro
// 7 = Ultimaker
// 71 = Ultimaker (Older electronics. Pre 1.5.4. This is rare)
// 72 = Ultimainboard 2.x (Uses TEMP_SENSOR 20)
// 77 = 3Drag Controller
// 8 = Teensylu
// 80 = Rumba
// 81 = Printrboard (AT90USB1286)
// 82 = Brainwave (AT90USB646)
// 83 = SAV Mk-I (AT90USB1286)
// 9 = Gen3+
// 70 = Megatronics
// 701= Megatronics v2.0

```

```

// 702= Minitronics v1.0
// 90 = Alpha OMCA board
// 91 = Final OMCA board
// 301= Rambo
// 21 = Elefu Ra Board (v3)

#ifndef MOTHERBOARD
#define MOTHERBOARD 33 //此处33为RAMPS 1.4扩展板类型
#endif

//液晶屏显示的名字, 不支持中文, 此处作为新版本的版本号显示
#define CUSTOM_MENDEL_NAME "UM v2.4"

// 定义挤出头的数量
#define EXTRUDERS 1

//// 电源电压电流
#define POWER_SUPPLY 1

//=====
//=====MICROMAKE 3D打印机 三角洲结构配置=====
//=====
#define DELTA

// //减小这个数值, 来缓解卡顿现象, 如修改为120进行测试。
#define DELTA_SEGMENTS_PER_SECOND 160

// 碳杆长度, 从一端球中心到另一端球中心的距离 大小调整此参数
#define DELTA_DIAGONAL_ROD 214// mm

// 打印头到滑杆水平距离 凹凸调整此参数
#define DELTA_SMOOTH_ROD_OFFSET 160// mm

// 效应器球中心和打印头的水平距离
#define DELTA_EFFECTOR_OFFSET 24.0 // mm

// 滑车球中心到滑杆水平距离
#define DELTA_CARRIAGE_OFFSET 22.0 // mm

// 三角洲半径。(打印头到滑杆水平距离-效应器球中心和打印头的水平距离-滑车球中心到滑杆水平距离)
#define DELTA_RADIUS (DELTA_SMOOTH_ROD_OFFSET-DELTA_EFFECTOR_OFFSET-
    DELTA_CARRIAGE_OFFSET)

// 打印半径
#define DELTA_PRINTABLE_RADIUS 90.0

#define SIN_60 0.8660254037844386
#define COS_60 0.5
#define DELTA_TOWER1_X -SIN_60*DELTA_RADIUS
#define DELTA_TOWER1_Y -COS_60*DELTA_RADIUS
#define DELTA_TOWER2_X SIN_60*DELTA_RADIUS
#define DELTA_TOWER2_Y -COS_60*DELTA_RADIUS
#define DELTA_TOWER3_X 0.0
#define DELTA_TOWER3_Y DELTA_RADIUS

#define DELTA_DIAGONAL_ROD_2 pow(DELTA_DIAGONAL_ROD,2)

```

```
//=====
//=====MICROMAKE 3D打印机 传感器设置=====
//=====

//配置传感器，根据挤出机个数来配置连接传感器数量。如果只有1个挤出机，则只需要开启传感器0接口和
    热床接口即可。
#define TEMP_SENSOR_0 5 //设置传感器0接口连接的传感器类型编号，类型根据上面说明设置相应的
    编号
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0 //设置传感器2接口连接的传感器编号，0表示关闭该端口
#define TEMP_SENSOR_BED 5 //设置热床传感器端口连接的传感器类型。该项如果设置错误将影响加
    热床温度控制
//添加热床支持只需将#define TEMP_SENSOR_BED 处0设置为5即可

//这里用传感器1来做传感器0的冗余。如果两个传感器温度差较大，将停止打印。
//#define TEMP_SENSOR_1_AS_REDUNDANT //设置传感器1作为冗余传感器。
#define MAX_REDUNDANT_TEMP_SENSOR_DIFF 10, //设置温度最大差值

// 打印之前通过M109检查当前温度已经接近设置温度，并等待N秒作为缓冲。
#define TEMP_RESIDENCY_TIME 10 // 设置达到设置温度后等待时间，单位秒
#define TEMP_HYSTERESIS 3 //设置离设置温度的浮动范围
#define TEMP_WINDOW 1

//最低温度低于N时，加热头将不会工作。该功能确保温度传感器连接或配置错误时不会烧毁设备。
//检查热敏电阻是否正常。
//如果热敏电阻工作不正常，将使加热头电源一直工作。这是非常危险的。

#define HEATER_0_MINTEMP 5 //设置加热头0的最小温度，一般设置成室内最低温度比较好。因为开
    机时应该测量到的是室温。
#define HEATER_1_MINTEMP 5
#define HEATER_2_MINTEMP 5
#define BED_MINTEMP 5

//当温度超过最大设置值，加热头会自动关闭。
//该项配置是为了保护你的设备，避免加热温度过高产生以外。但不能防止温度传感器非正常工作的情况。
//你应该使用MINTEMP选项来保证温度传感器短路或损坏时的设备安全。
#define HEATER_0_MAXTEMP 275 //挤出头0 最大保护温度
#define HEATER_1_MAXTEMP 275
#define HEATER_2_MAXTEMP 275
#define BED_MAXTEMP 120 //热床最大保护温度

//如果你的热床电流较大，你可以通过设置占空比的方式降低电流，这个值应该是个整数，数字越大，电流
    越小。
//#define HEATER_BED_DUTY_CYCLE_DIVIDER 4

//如果你想用M105命令来显示加热器的功耗，需要设置下面两个参数
//#define EXTRUDER_WATTS (12.0*12.0/6.7) // P=I^2/R
//#define BED_WATTS (12.0*12.0/1.1) // P=I^2/R

//PID设置
#define PIDTEMP
#define BANG_MAX 255
```

```

#define PID_MAX 255
#ifndef PIDTEMP
  #define PID_FUNCTIONAL_RANGE 10
  #define PID_INTEGRAL_DRIVE_MAX 255
  #define K1 0.95
  #define PID_dT ((OVERSAMPLER * 8.0)/(F_CPU / 64.0 / 256.0))
  #define DEFAULT_Kp 22.2
  #define DEFAULT_Ki 1.08
  #define DEFAULT_Kd 114
#endif

#define MAX_BED_POWER 255 //通过占空比方式限制热床的最大功率，255表示不限制
#ifndef PIDTEMPBED
  #define DEFAULT_bedKp 10.00
  #define DEFAULT_bedKi .023
  #define DEFAULT_bedKd 305.4
#endif

//为了防止加热头未开启时的冷挤出，这里设置当加热头温度未达到N时不允许挤出操作执行。（M302指令
//可以解除冷挤出限制）
#define PREVENT_DANGEROUS_EXTRUDE
#define PREVENT_LENGTHY_EXTRUDE

#define EXTRUDE_MINTEMP 175//设置挤出头运行的最低温度
#define EXTRUDE_MAXLENGTH (X_MAX_LENGTH+Y_MAX_LENGTH) //避免非常长的挤出操作

//=====
//===== 机械方面设置 =====
//=====

// #define COREXY //取消前面的注释可以期待用corexy运动系统

// 限位开关设置
#define ENDSTOPPULLUPS //将上面参数用“//”注释掉，将禁用限位开关的上拉电阻。该配置是全局
//配置，不用该参数可以用下面单独设置是否开启上拉电阻

#ifndef ENDSTOPPULLUPS
  //分别对限位开关单独设置上拉电阻。如果ENDSTOPPULLUPS被定义，该配置将被忽略
  // #define ENDSTOPPULLUP_XMAX
  // #define ENDSTOPPULLUP_YMAX
  // #define ENDSTOPPULLUP_ZMAX
  // #define ENDSTOPPULLUP_XMIN
  // #define ENDSTOPPULLUP_YMIN
  // #define ENDSTOPPULLUP_ZMIN
#endif

#ifndef ENDSTOPPULLUPS
  #define ENDSTOPPULLUP_XMAX
  #define ENDSTOPPULLUP_YMAX
  #define ENDSTOPPULLUP_ZMAX
  #define ENDSTOPPULLUP_XMIN
  #define ENDSTOPPULLUP_YMIN
  #define ENDSTOPPULLUP_ZMIN
#endif

//如果你使用机械式的限位开关，并且接到了信号和GND两个接口，那么上面的上拉配置需要打开
//配置3个轴的限位开关类型的，配置为true，限位开关应该接常开端子。如果你接常闭端子，则将true

```

改为false

```
//设置为true来颠倒限位开关逻辑值。如果设置为true时，限位开关实际的开/合与检测相反，则将该参数配置为false
const bool X_MIN_ENDSTOP_INVERTING = false;
const bool Y_MIN_ENDSTOP_INVERTING = false;
const bool Z_MIN_ENDSTOP_INVERTING = false;
const bool X_MAX_ENDSTOP_INVERTING = false;
const bool Y_MAX_ENDSTOP_INVERTING = false;
const bool Z_MAX_ENDSTOP_INVERTING = false;
//#define DISABLE_MAX_ENDSTOPS
//#define DISABLE_MIN_ENDSTOPS

//为了挡块检查程序的兼容性禁用最大终点挡块
#if defined(COREXY) && !defined(DISABLE_MAX_ENDSTOPS)
  #define DISABLE_MAX_ENDSTOPS
#endif

//设置步进电机使能引脚的电平。（4988模块保持0即可）
#define X_ENABLE_ON 0
#define Y_ENABLE_ON 0
#define Z_ENABLE_ON 0
#define E_ENABLE_ON 0 // 针对所有挤出机有效

//当哪个轴不运动时是否关闭电机。（注意：如果这里打开将会使电机在不使用时被锁止，而导致电机温度急剧上升）
#define DISABLE_X false
#define DISABLE_Y false
#define DISABLE_Z false
#define DISABLE_E false //针对所有挤出机有效

//电机运动方向控制。由于电机连线不同，电机的运动方向也不同，但打印机的0点位置在左下角，如果电机的运动方向
//与控制方向不同，则可以将下面参数值true和false对调，也可以将步进电机的4根线反过来插。
#define INVERT_X_DIR true //（X轴配置）
#define INVERT_Y_DIR true //（Y轴配置）
#define INVERT_Z_DIR true //（Z轴配置）
#define INVERT_E0_DIR false //（挤出机0配置）
#define INVERT_E1_DIR false //（挤出机1配置）
#define INVERT_E2_DIR false //（挤出机2配置）

//停止开关设置
//设置回0时，电机的运动方向。1最大限位方向，-1最小限位方向。一般都是设置为-1
#define X_HOME_DIR 1
#define Y_HOME_DIR 1
#define Z_HOME_DIR 1
//软限位开关设置
#define min_software_endstops false //最小值设置，如果设置为true，则移动距离
<HOME_POS值
#define max_software_endstops true //最大值设置，如果设置为true，轴不会移动到坐标大于
下面定义的长度。

//各轴的软件限位值
#define X_MAX_POS DELTA_PRINTABLE_RADIUS
#define X_MIN_POS -DELTA_PRINTABLE_RADIUS
```

```

#define Y_MAX_POS DELTA_PRINTABLE_RADIUS
#define Y_MIN_POS -DELTA_PRINTABLE_RADIUS
#define Z_MAX_POS MANUAL_Z_HOME_POS
#define Z_MIN_POS 0

#define X_MAX_LENGTH (X_MAX_POS - X_MIN_POS)
#define Y_MAX_LENGTH (Y_MAX_POS - Y_MIN_POS)
#define Z_MAX_LENGTH (Z_MAX_POS - Z_MIN_POS)
//===== MICROMAKE 3D打印机 自动调平配置 =====

#define ENABLE_AUTO_BED_LEVELING // 是否开启自动调平功能

#ifndef ENABLE_AUTO_BED_LEVELING

#define DELTA_PROBABLE_RADIUS (DELTA_PRINTABLE_RADIUS-50)//此处设置为调平探针移动范围，增大调平范围减少“-50”这个值，减少调平范围增大“-50”这个值
//如: #define DELTA_PROBABLE_RADIUS (DELTA_PRINTABLE_RADIUS-60)

#define LEFT_PROBE_BED_POSITION -DELTA_PROBABLE_RADIUS
#define RIGHT_PROBE_BED_POSITION DELTA_PROBABLE_RADIUS
#define BACK_PROBE_BED_POSITION DELTA_PROBABLE_RADIUS
#define FRONT_PROBE_BED_POSITION -DELTA_PROBABLE_RADIUS

#define X_PROBE_OFFSET_FROM_EXTRUDER 0.0
#define Y_PROBE_OFFSET_FROM_EXTRUDER 0.0
#define Z_PROBE_OFFSET_FROM_EXTRUDER 1.6//自动调平设置 过高减小 过低增大

#define Z_RAISE_BEFORE_HOMING 4 // 配置回原点前Z轴升起的高度，该高度要确保在Z轴最大高度范围内。

#define XY_TRAVEL_SPEED 2000 //执行自动调平移动的速度，增大速度增加，减小速度降低

#define Z_RAISE_BEFORE_PROBING 80 //经过第一个检测点前Z轴抬起的高度，该高度要确保调平传感器可以正常放下。
#define Z_RAISE_BETWEEN_PROBINGS 5 //经过下一个检测点前Z轴抬起的高度

#define Z_SAFE_HOMING
#ifndef Z_SAFE_HOMING

#define Z_SAFE_HOMING_X_POINT (X_MAX_LENGTH/2)
#define Z_SAFE_HOMING_Y_POINT (Y_MAX_LENGTH/2)

#endif

#define ACCURATE_BED_LEVELING
#ifndef ACCURATE_BED_LEVELING
#define ACCURATE_BED_LEVELING_POINTS 3 //自动调平探头点点数 3为横竖向各点3个点，共9点，改为6就是横竖向各点6个点，共36个点。
#define ACCURATE_BED_LEVELING_GRID_X ((RIGHT_PROBE_BED_POSITION - LEFT_PROBE_BED_POSITION) / (ACCURATE_BED_LEVELING_POINTS - 1))
#define ACCURATE_BED_LEVELING_GRID_Y ((BACK_PROBE_BED_POSITION - FRONT_PROBE_BED_POSITION) / (ACCURATE_BED_LEVELING_POINTS - 1))
#define NONLINEAR_BED_LEVELING

```

```

    #endif

#endif

//归位开关设置
#define MANUAL_HOME_POSITIONS //如果开启该配置, 下面 MANUAL_*_HOME_POS配置将生效
#define BED_CENTER_AT_0_0 //如果开启该配置, 热床的中心位置在(X=0, Y=0)

//手动回零开关的位置:
//对于三角洲结构这意味着笛卡尔打印机的顶部和中心的值。
#define MANUAL_X_HOME_POS 0
#define MANUAL_Y_HOME_POS 0
#define MANUAL_Z_HOME_POS 230//306.6 // Z轴高度设置
//因每台机器安装会有差别, 需自行测量, 测量方法请查看配套视频教程, 设置完后后记得保持修改

//轴设置
#define NUM_AXIS 4 //轴的数量, 各轴的配置是顺序是X, Y, Z, E
#define HOMING_FEEDRATE {60*60, 60*60, 60*60, 0} //配置归位时的速度

#define XYZ_FULL_STEPS_PER_ROTATION 200 //步进电机每周的步数, 即360/步进电机上的角度
//如1.8度, 步数应该是360/1.8=200; 如果是0.9度电机的话就是 360/0.9=400。27号以前购买的用
    户请修改为400, 27号以后的用户请修改为200。
#define XYZ_MICROSTEPS 16 //步进驱动的细分数
#define XYZ_BELT_PITCH 2 //同步带齿间距
#define XYZ_PULLEY_TEETH 16 //同步轮齿数
#define XYZ_STEPS (XYZ_FULL_STEPS_PER_ROTATION * XYZ_MICROSTEPS / double
    (XYZ_BELT_PITCH) / double(XYZ_PULLEY_TEETH))
//这是计算公式: 步进电机数*步进驱动的细分数/同步带齿间距/同步轮齿数

#define DEFAULT_AXIS_STEPS_PER_UNIT {XYZ_STEPS, XYZ_STEPS, XYZ_STEPS,
    98} //挤出机挤出量
#define DEFAULT_MAX_FEEDRATE {200, 200, 200, 200}
#define DEFAULT_MAX_ACCELERATION {3000, 3000, 3000, 3000}

//加速度配置, 假如打印时失步太大, 可以将这个值改小
#define DEFAULT_ACCELERATION 3000
#define DEFAULT_RETRACT_ACCELERATION 3000

//各轴不需要加速的距离, 即无需加速, 立即完成的距离 (即软件认为他可以在瞬间完成的)
#define DEFAULT_XYJERK 20.0
#define DEFAULT_ZJERK 20.0
#define DEFAULT_EJERK 20.0

//=====
//=====附加功能=====
//=====
//以下内容暂未汉化, 后续会持续汉化更新, 欢迎加入我们的交流QQ群: 224981313
//感谢您的支持, 也欢迎更多朋友可以持续补充, 完善, 欢迎散播复制, 尊重劳动者, 使用发布请注明出
    处。
// EEPROM
// The microcontroller can store settings in the EEPROM, e.g. max velocity...
// M500 - stores parameters in EEPROM
// M501 - reads parameters from EEPROM (if you need reset them after you
    changed them temporarily).

```

```

// M502 – reverts to the default "factory settings". You still need to store
// them in EEPROM afterwards if you want to.
//define this to enable EEPROM support
//#define EEPROM_SETTINGS
//to disable EEPROM Serial responses and decrease program space by ~1700 byte:
// comment this out:
// please keep turned on if you can.
#define EEPROM_CHITCHAT

// Preheat Constants
#define PLA_PREHEAT_HOTEND_TEMP 180
#define PLA_PREHEAT_HPB_TEMP 70
#define PLA_PREHEAT_FAN_SPEED 255 // Insert Value between 0 and 255

#define ABS_PREHEAT_HOTEND_TEMP 240
#define ABS_PREHEAT_HPB_TEMP 100
#define ABS_PREHEAT_FAN_SPEED 255 // Insert Value between 0 and 255

//LCD and SD support
//#define ULTRA_LCD //general LCD support, also 16x2
//#define DOGLCD // Support for SPI LCD 128x64 (Controller ST7565R graphic
// Display Family)
//#define SDSUPPORT // Enable SD Card Support in Hardware Console
//#define SDSLOW // Use slower SD transfer mode (not normally needed –
// uncomment if you're getting volume init error)
//#define ENCODER_PULSES_PER_STEP 1 // Increase if you have a high resolution
// encoder
//#define ENCODER_STEPS_PER_MENU_ITEM 5 // Set according to
// ENCODER_PULSES_PER_STEP or your liking
//#define ULTIMAKERCONTROLLER //as available from the Ultimaker online store.
//#define ULTIPANEL //the UltiPanel as on Thingiverse
//#define LCD_FEEDBACK_FREQUENCY_HZ 1000 // this is the tone frequency the
// buzzer plays when on UI feedback. ie Screen Click
//#define LCD_FEEDBACK_FREQUENCY_DURATION_MS 100 // the duration the buzzer
// plays the UI feedback sound. ie Screen Click

// The MaKr3d Makr-Panel with graphic controller and SD support
// http://reprap.org/wiki/MaKr3d\_MaKrPanel
//#define MAKRPANEL

// The RepRapDiscount Smart Controller (white PCB)
// http://reprap.org/wiki/RepRapDiscount\_Smart\_Controller
#define REPRAP_DISCOUNT_SMART_CONTROLLER

// The GADGETS3D G3D LCD/SD Controller (blue PCB)
// http://reprap.org/wiki/RAMPS\_1.3/1.4\_GADGETS3D\_Shield\_with\_Panel
//#define G3D_PANEL

// The RepRapDiscount FULL GRAPHIC Smart Controller (quadratic white PCB)
// http://reprap.org/wiki/RepRapDiscount\_Full\_Graphic\_Smart\_Controller
//
// ==> REMEMBER TO INSTALL U8glib to your ARDUINO library folder: http://code.google.com/p/u8glib/wiki/u8glib
//#define REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER

// The RepRapWorld REPRAPWORLD_KEYPAD v1.1
// http://reprapworld.com/?products\_details&products\_id=202&cPath=1591\_1626
//#define REPRAPWORLD_KEYPAD
//#define REPRAPWORLD_KEYPAD_MOVE_STEP 10.0 // how much should be moved when a
// key is pressed, eg 10.0 means 10mm per click

```



```

// The Elefu RA Board Control Panel
// http://www.elefu.com/index.php?route=product/product&product\_id=53
// REMEMBER TO INSTALL LiquidCrystal_I2C.h in your ARUDINO library folder:
// https://github.com/kiyoshigawa/LiquidCrystal\_I2C
//#define RA_CONTROL_PANEL

//automatic expansion
#if defined (MAKRPANEL)
  #define DOGLCD
  #define SDSUPPORT
  #define ULTIPANEL
  #define NEWPANEL
  #define DEFAULT_LCD_CONTRAST 17
#endif

#if defined (REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER)
  #define DOGLCD
  #define U8GLIB_ST7920
  #define REPRAP_DISCOUNT_SMART_CONTROLLER
#endif

#if defined(ULTIMAKERCONTROLLER) || defined(REPRAP_DISCOUNT_SMART_CONTROLLER)
  || defined(G3D_PANEL)
  #define ULTIPANEL
  #define NEWPANEL
#endif

#if defined(REPRAPWORLD_KEYPAD)
  #define NEWPANEL
  #define ULTIPANEL
#endif
#if defined(RA_CONTROL_PANEL)
  #define ULTIPANEL
  #define NEWPANEL
  #define LCD_I2C_TYPE_PCA8574
  #define LCD_I2C_ADDRESS 0x27 // I2C Address of the port expander
#endif

//I2C PANELS

//#define LCD_I2C_SAINSMART_YWROBOT
#ifndef LCD_I2C_SAINSMART_YWROBOT
  // This uses the LiquidCrystal_I2C library ( https://bitbucket.org/fmalpartida/new-liquidcrystal/wiki/Home )
  // Make sure it is placed in the Arduino libraries directory.
  #define LCD_I2C_TYPE_PCF8575
  #define LCD_I2C_ADDRESS 0x27 // I2C Address of the port expander
  #define NEWPANEL
  #define ULTIPANEL
#endif

// PANEL0LU2 LCD with status LEDs, separate encoder and click inputs
//#define LCD_I2C_PANEL0LU2
#ifndef LCD_I2C_PANEL0LU2
  // This uses the LiquidTWI2 library v1.2.3 or later ( https://github.com/lincomatic/LiquidTWI2 )
  // Make sure the LiquidTWI2 directory is placed in the Arduino or Sketchbook
  libraries subdirectory.
  // (v1.2.3 no longer requires you to define PANEL0LU in the LiquidTWI2.h

```

```

    library header file)
// Note: The PANEL0LU2 encoder click input can either be directly connected
    to a pin
//      (if BTN_ENC defined to != -1) or read through I2C (when BTN_ENC ==
    -1).
#define LCD_I2C_TYPE_MCP23017
#define LCD_I2C_ADDRESS 0x20 // I2C Address of the port expander
#define LCD_USE_I2C_BUZZER //comment out to disable buzzer on LCD
#define NEWPANEL
#define ULTIPANEL

#ifndef ENCODER_PULSES_PER_STEP
    #define ENCODER_PULSES_PER_STEP 4
#endif

#ifndef ENCODER_STEPS_PER_MENU_ITEM
    #define ENCODER_STEPS_PER_MENU_ITEM 1
#endif

#ifdef LCD_USE_I2C_BUZZER
    #define LCD_FEEDBACK_FREQUENCY_HZ 1000
    #define LCD_FEEDBACK_FREQUENCY_DURATION_MS 100
#endif

#endif

// Panucatt VIKI LCD with status LEDs, integrated click & L/R/U/P buttons,
    separate encoder inputs
// #define LCD_I2C_VIKI
#ifdef LCD_I2C_VIKI
    // This uses the LiquidTWI2 library v1.2.3 or later ( https://github.com/lincomatic/LiquidTWI2 )
    // Make sure the LiquidTWI2 directory is placed in the Arduino or Sketchbook
        libraries subdirectory.
    // Note: The pause/stop/resume LCD button pin should be connected to the
        Arduino
    //      BTN_ENC pin (or set BTN_ENC to -1 if not used)
#define LCD_I2C_TYPE_MCP23017
#define LCD_I2C_ADDRESS 0x20 // I2C Address of the port expander
#define LCD_USE_I2C_BUZZER //comment out to disable buzzer on LCD (requires
        LiquidTWI2 v1.2.3 or later)
#define NEWPANEL
#define ULTIPANEL
#endif

// Shift register panels
// -----
// 2 wire Non-latching LCD SR from:
// https://bitbucket.org/fmalpartida/new-liquidcrystal/wiki/schematics#!
// shiftregister-connection
// #define SR_LCD
#ifdef SR_LCD
    #define SR_LCD_2W_NL // Non latching 2 wire shift register
    // #define NEWPANEL
#endif

#ifdef ULTIPANEL
    // #define NEWPANEL //enable this if you have a click-encoder panel

```

```

#define SDSUPPORT
#define ULTRA_LCD
#ifndef DOGLCD // Change number of lines to match the DOG graphic display
  #define LCD_WIDTH 20
  #define LCD_HEIGHT 5
#else
  #define LCD_WIDTH 20
  #define LCD_HEIGHT 4
#endif
#else //no panel but just LCD
  #ifndef ULTRA_LCD
  #ifndef DOGLCD // Change number of lines to match the 128x64 graphics display
    #define LCD_WIDTH 20
    #define LCD_HEIGHT 5
  #else
    #define LCD_WIDTH 16
    #define LCD_HEIGHT 2
  #endif
  #endif
#endif

// default LCD contrast for dogm-like LCD displays
#ifndef DOGLCD
# ifndef DEFAULT_LCD_CONTRAST
#  define DEFAULT_LCD_CONTRAST 32
# endif
#endif

// Increase the FAN pwm frequency. Removes the PWM noise but increases heating
  in the FET/Arduino
//#define FAST_PWM_FAN

// Temperature status LEDs that display the hotend and bed temperature.
// If all hotends and bed temperature and temperature setpoint are < 54C then
  the BLUE led is on.
// Otherwise the RED led is on. There is 1C hysteresis.
//#define TEMP_STAT_LEDS

// Use software PWM to drive the fan, as for the heaters. This uses a very low
  frequency
// which is not ass annoying as with the hardware PWM. On the other hand, if
  this frequency
// is too low, you should also increment SOFT_PWM_SCALE.
//#define FAN_SOFT_PWM

// Incrementing this by 1 will double the software PWM frequency,
// affecting heaters, and the fan if FAN_SOFT_PWM is enabled.
// However, control resolution will be halved for each increment;
// at zero value, there are 128 effective control positions.
#define SOFT_PWM_SCALE 0

// M240 Triggers a camera by emulating a Canon RC-1 Remote
// Data from: http://www.doc-diy.net/photo/rc-1\_hacked/
// #define PHOTOGRAPH_PIN 23

// SF send wrong arc g-codes when using Arc Point as fillet procedure
//#define SF_ARC_FIX

// Support for the BariCUDA Paste Extruder.
//#define BARICUDA

```

```
//define BlinkM/CyzRgb Support
//#define BLINKM

/*****\
* R/C SERVO support
* Sponsored by TrinityLabs, Reworked by codexmas
*****/

// Number of servos
//
// If you select a configuration below, this will receive a default value and
// does not need to be set manually
// set it manually if you have more servos than extruders and wish to manually
// control some
// leaving it undefined or defining as 0 will disable the servo subsystem
// If unsure, leave commented / disabled
//
//#define NUM_SERVOS 3 // Servo index starts with 0 for M280 command

// Servo Endstops
//
// This allows for servo actuated endstops, primary usage is for the Z Axis to
// eliminate calibration or bed height changes.
// Use M206 command to correct for switch height offset to actual nozzle
// height. Store that setting with M500.
//
//#define SERVO_ENDSTOPS {-1, -1, 0} // Servo index for X, Y, Z. Disable with
-1
//#define SERVO_ENDSTOP_ANGLES {0,0, 0,0, 70,0} // X,Y,Z Axis Extend and
Retract angles

#include "Configuration_adv.h"
#include "thermistortables.h"

#endif //__CONFIGURATION_H
```