my Config file for mkbase with bltouch

Ken Stewart·Thursday, May 19, 2016

DO NOT JUST COPY AND COMPILE THIS TO YOUR BOARD WITHOUT FIRST LOOKING OVER IT AND CHANGING THING TO YOUR SETTING THAT ARE SPECIFIC TO YOUR PRINT. mine is not a stock tevo it has a 500w silicon heated bed, a E3D volcano nozzle, BLTouch and a air tripper extruder. this is only te be used as an example to check over the settings of the BLtouch and even those setting for the offset need to be set up for YOUR PRINTER

USE AT YOUR OWN RISK

#ifndef CONFIGURATION\_H

#define CONFIGURATION\_H

#include "boards.h"

// This configuration file contains the basic settings.

// Advanced settings can be found in Configuration\_adv.h

// BASIC SETTINGS: select your board type, temperature sensor type, axis scaling, and endstop configuration

//===========================================================================

//============================= DELTA Printer ===============================

//===========================================================================

// For a Delta printer replace the configuration files with the files in the

// example\_configurations/delta directory.

//

//===========================================================================

//============================= SCARA Printer ===============================

//===========================================================================

// For a Delta printer replace the configuration files with the files in the

// example\_configurations/SCARA directory.

//

// User-specified version info of this build to display in [Pronterface, etc] terminal window during

// startup. Implementation of an idea by Prof Braino to inform user that any changes made to this

// build by the user have been successfully uploaded into firmware.

#define STRING\_VERSION\_CONFIG\_H \_\_DATE\_\_ " " \_\_TIME\_\_ // build date and time

#define STRING\_CONFIG\_H\_AUTHOR "Kat blt-shv" // Who made the changes.

// SERIAL\_PORT selects which serial port should be used for communication with the host.

// This allows the connection of wireless adapters (for instance) to non-default port pins.

// Serial port 0 is still used by the Arduino bootloader regardless of this setting.

#define SERIAL\_PORT 0

// This determines the communication speed of the printer

#define BAUDRATE 250000

// This enables the serial port associated to the Bluetooth interface

//#define BTENABLED // Enable BT interface on AT90USB devices

// The following define selects which electronics board you have.

// Please choose the name from boards.h that matches your setup

#ifndef MOTHERBOARD

#define MOTHERBOARD 33

// #define MOTHERBOARD BOARD\_ULTIMAKER

#endif

// Define this to set a custom name for your generic Mendel,

#define CUSTOM\_MENDEL\_NAME "Tarantula"

// Define this to set a unique identifier for this printer, (Used by some programs to differentiate between machines)

// You can use an online service to generate a random UUID. (eg http://www.uuidgenerator.net/versio...)

#define MACHINE\_UUID "0d4aecf0-544e-11e5-885d-feff819cdc9f"

// This defines the number of extruders

#define EXTRUDERS 1

//// The following define selects which power supply you have. Please choose the one that matches your setup

// 1 = ATX

// 2 = X-Box 360 203Watts (the blue wire connected to PS\_ON and the red wire to VCC)

#define POWER\_SUPPLY 1

// Define this to have the electronics keep the power supply off on startup. If you don't know what this is leave it.

// #define PS\_DEFAULT\_OFF

//===========================================================================

//=============================Thermal Settings ============================

//===========================================================================

//

//--NORMAL IS 4.7kohm PULLUP!-- 1kohm pullup can be used on hotend sensor, using correct resistor and table

//

//// Temperature sensor settings:

// -2 is thermocouple with MAX6675 (only for sensor 0)

// -1 is thermocouple with AD595

// 0 is not used

// 1 is 100k thermistor - best choice for EPCOS 100k (4.7k pullup)

// 2 is 200k thermistor - ATC Semitec 204GT-2 (4.7k pullup)

// 3 is Mendel-parts thermistor (4.7k pullup)

// 4 is 10k thermistor !! do not use it for a hotend. It gives bad resolution at high temp. !!

// 5 is 100K thermistor - ATC Semitec 104GT-2 (Used in ParCan & J-Head) (4.7k pullup)

// 6 is 100k EPCOS - Not as accurate as table 1 (created using a fluke thermocouple) (4.7k pullup)

// 7 is 100k Honeywell thermistor 135-104LAG-J01 (4.7k pullup)

// 71 is 100k Honeywell thermistor 135-104LAF-J01 (4.7k pullup)

// 8 is 100k 0603 SMD Vishay NTCS0603E3104FXT (4.7k pullup)

// 9 is 100k GE Sensing AL03006-58.2K-97-G1 (4.7k pullup)

// 10 is 100k RS thermistor 198-961 (4.7k pullup)

// 11 is 100k beta 3950 1% thermistor (4.7k pullup)

// 12 is 100k 0603 SMD Vishay NTCS0603E3104FXT (4.7k pullup) (calibrated for Makibox hot bed)

// 13 is 100k Hisens 3950 1% up to 300°C for hotend "Simple ONE " & "Hotend "All In ONE"

// 20 is the PT100 circuit found in the Ultimainboard V2.x

// 60 is 100k Maker's Tool Works Kapton Bed Thermistor beta=3950

//

// 1k ohm pullup tables - This is not normal, you would have to have changed out your 4.7k for 1k

// (but gives greater accuracy and more stable PID)

// 51 is 100k thermistor - EPCOS (1k pullup)

// 52 is 200k thermistor - ATC Semitec 204GT-2 (1k pullup)

// 55 is 100k thermistor - ATC Semitec 104GT-2 (Used in ParCan & J-Head) (1k pullup)

//

// 1047 is Pt1000 with 4k7 pullup

// 1010 is Pt1000 with 1k pullup (non standard)

// 147 is Pt100 with 4k7 pullup

// 110 is Pt100 with 1k pullup (non standard)

#define TEMP\_SENSOR\_0 1

#define TEMP\_SENSOR\_1 0

#define TEMP\_SENSOR\_2 0

#define TEMP\_SENSOR\_BED 1

// This makes temp sensor 1 a redundant sensor for sensor 0. If the temperatures difference between these sensors is to high the print will be aborted.

//#define TEMP\_SENSOR\_1\_AS\_REDUNDANT

#define MAX\_REDUNDANT\_TEMP\_SENSOR\_DIFF 10

// Actual temperature must be close to target for this long before M109 returns success

#define TEMP\_RESIDENCY\_TIME 5 // (seconds)

#define TEMP\_HYSTERESIS 3 // (degC) range of +/- temperatures considered "close" to the target one

#define TEMP\_WINDOW 1 // (degC) Window around target to start the residency timer x degC early.

// The minimal temperature defines the temperature below which the heater will not be enabled It is used

// to check that the wiring to the thermistor is not broken.

// Otherwise this would lead to the heater being powered on all the time.

#define HEATER\_0\_MINTEMP 5

#define HEATER\_1\_MINTEMP 5

#define HEATER\_2\_MINTEMP 5

#define BED\_MINTEMP 5

// When temperature exceeds max temp, your heater will be switched off.

// This feature exists to protect your hotend from overheating accidentally, but \*NOT\* from thermistor short/failure!

// You should use MINTEMP for thermistor short/failure protection.

#define HEATER\_0\_MAXTEMP 280

#define HEATER\_1\_MAXTEMP 275

#define HEATER\_2\_MAXTEMP 275

#define BED\_MAXTEMP 130

// If your bed has low resistance e.g. .6 ohm and throws the fuse you can duty cycle it to reduce the

// average current. The value should be an integer and the heat bed will be turned on for 1 interval of

// HEATER\_BED\_DUTY\_CYCLE\_DIVIDER intervals.

//#define HEATER\_BED\_DUTY\_CYCLE\_DIVIDER 4

// If you want the M105 heater power reported in watts, define the BED\_WATTS, and (shared for all extruders) EXTRUDER\_WATTS

//#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 settings:

// Comment the following line to disable PID and enable bang-bang.

#define PIDTEMP

#define BANG\_MAX 255 // limits current to nozzle while in bang-bang mode; 255=full current

#define PID\_MAX BANG\_MAX // limits current to nozzle while PID is active (see PID\_FUNCTIONAL\_RANGE below); 255=full current

#ifdef PIDTEMP

//#define PID\_DEBUG // Sends debug data to the serial port.

//#define PID\_OPENLOOP 1 // Puts PID in open loop. M104/M140 sets the output power from 0 to PID\_MAX

#define PID\_FUNCTIONAL\_RANGE 10 // If the temperature difference between the target temperature and the actual temperature

// is more then PID\_FUNCTIONAL\_RANGE then the PID will be shut off and the heater will be set to min/max.

#define PID\_INTEGRAL\_DRIVE\_MAX PID\_MAX //limit for the integral term

#define K1 0.95 //smoothing factor within the PID

#define PID\_dT ((OVERSAMPLENR \* 10.0)/(F\_CPU / 64.0 / 256.0)) //sampling period of the temperature routine

// If you are using a pre-configured hotend then you can use one of the value sets by uncommenting it

// TEVO Tarantula Stock PID Settings - Hotend

// #define DEFAULT\_Kp 22.2

// #define DEFAULT\_Ki 1.08

// #define DEFAULT\_Kd 114

// TEVO Tarantula Custom PID Settings - Hotend

#define DEFAULT\_Kp 16.10

#define DEFAULT\_Ki 0.76

#define DEFAULT\_Kd 85.73

// MakerGear

// #define DEFAULT\_Kp 7.0

// #define DEFAULT\_Ki 0.1

// #define DEFAULT\_Kd 12

// Mendel Parts V9 on 12V

// #define DEFAULT\_Kp 63.0

// #define DEFAULT\_Ki 2.25

// #define DEFAULT\_Kd 440

#endif // PIDTEMP

// Bed Temperature Control

// Select PID or bang-bang with PIDTEMPBED. If bang-bang, BED\_LIMIT\_SWITCHING will enable hysteresis

//

// Uncomment this to enable PID on the bed. It uses the same frequency PWM as the extruder.

// If your PID\_dT above is the default, and correct for your hardware/configuration, that means 7.689Hz,

// which is fine for driving a square wave into a resistive load and does not significantly impact you FET heating.

// This also works fine on a Fotek SSR-10DA Solid State Relay into a 250W heater.

// If your configuration is significantly different than this and you don't understand the issues involved, you probably

// shouldn't use bed PID until someone else verifies your hardware works.

// If this is enabled, find your own PID constants below.

#define PIDTEMPBED

//

//#define BED\_LIMIT\_SWITCHING

// This sets the max power delivered to the bed, and replaces the HEATER\_BED\_DUTY\_CYCLE\_DIVIDER option.

// all forms of bed control obey this (PID, bang-bang, bang-bang with hysteresis)

// setting this to anything other than 255 enables a form of PWM to the bed just like HEATER\_BED\_DUTY\_CYCLE\_DIVIDER did,

// so you shouldn't use it unless you are OK with PWM on your bed. (see the comment on enabling PIDTEMPBED)

#define MAX\_BED\_POWER 255 // limits duty cycle to bed; 255=full current

#ifdef PIDTEMPBED

//120v 250W silicone heater into 4mm borosilicate (MendelMax 1.5+)

//from FOPDT model - kp=.39 Tp=405 Tdead=66, Tc set to 79.2, aggressive factor of .15 (vs .1, 1, 10)

// TEVO Tarantula Stock PID Settings - Heatbed

// #define DEFAULT\_bedKp 13.64

// #define DEFAULT\_bedKi 0.68

// #define DEFAULT\_bedKd 68.72

// TEVO Tarantula Custom PID Settings

#define DEFAULT\_bedKp 450.06

#define DEFAULT\_bedKi 63.88

#define DEFAULT\_bedKd 792.67

//120v 250W silicone heater into 4mm borosilicate (MendelMax 1.5+)

//from pidautotune

// #define DEFAULT\_bedKp 97.1

// #define DEFAULT\_bedKi 1.41

// #define DEFAULT\_bedKd 1675.16

// FIND YOUR OWN: "M303 E-1 C8 S90" to run autotune on the bed at 90 degreesC for 8 cycles.

#endif // PIDTEMPBED

//this prevents dangerous Extruder moves, i.e. if the temperature is under the limit

//can be software-disabled for whatever purposes by

#define PREVENT\_DANGEROUS\_EXTRUDE

//if PREVENT\_DANGEROUS\_EXTRUDE is on, you can still disable (uncomment) very long bits of extrusion separately.

#define PREVENT\_LENGTHY\_EXTRUDE

// TEVO Tarantula Stock dangerous extrusion

//#define EXTRUDE\_MINTEMP 170

// TEVO Tarantula Custom dangerous extrusion

#define EXTRUDE\_MINTEMP 16

#define EXTRUDE\_MAXLENGTH (X\_MAX\_LENGTH+Y\_MAX\_LENGTH) //prevent extrusion of very large distances.

/\*================== Thermal Runaway Protection ==============================

This is a feature to protect your printer from burn up in flames if it has

a thermistor coming off place (this happened to a friend of mine recently and

motivated me writing this feature).

The issue: If a thermistor come off, it will read a lower temperature than actual.

The system will turn the heater on forever, burning up the filament and anything

else around.

After the temperature reaches the target for the first time, this feature will

start measuring for how long the current temperature stays below the target

minus \_HYSTERESIS (set\_temperature - THERMAL\_RUNAWAY\_PROTECTION\_HYSTERESIS).

If it stays longer than \_PERIOD, it means the thermistor temperature

cannot catch up with the target, so something \*may be\* wrong. Then, to be on the

safe side, the system will he halt.

Bear in mind the count down will just start AFTER the first time the

thermistor temperature is over the target, so you will have no problem if

your extruder heater takes 2 minutes to hit the target on heating.

\*/

// If you want to enable this feature for all your extruder heaters,

// uncomment the 2 defines below:

// Parameters for all extruder heaters

#define THERMAL\_RUNAWAY\_PROTECTION\_PERIOD 60 //in seconds

#define THERMAL\_RUNAWAY\_PROTECTION\_HYSTERESIS 10 // in degree Celsius

// If you want to enable this feature for your bed heater,

// uncomment the 2 defines below:

// Parameters for the bed heater

#define THERMAL\_RUNAWAY\_PROTECTION\_BED\_PERIOD 60 //in seconds

#define THERMAL\_RUNAWAY\_PROTECTION\_BED\_HYSTERESIS 10 // in degree Celsius

//===========================================================================

//===========================================================================

//=============================Mechanical Settings===========================

//===========================================================================

// Uncomment the following line to enable CoreXY kinematics

// #define COREXY

// coarse Endstop Settings

//#define ENDSTOPPULLUPS // Comment this out (using // at the start of the line) to disable the endstop pullup resistors

#ifndef ENDSTOPPULLUPS

// fine endstop settings: Individual pullups. will be ignored if ENDSTOPPULLUPS is defined

#define ENDSTOPPULLUP\_XMAX

#define ENDSTOPPULLUP\_YMAX

#define ENDSTOPPULLUP\_ZMAX

#define ENDSTOPPULLUP\_XMIN

#define ENDSTOPPULLUP\_YMIN

// #define ENDSTOPPULLUP\_ZMIN

#endif

// The pullups are needed if you directly connect a mechanical endswitch between the signal and ground pins.

const bool X\_MIN\_ENDSTOP\_INVERTING = true; // set to true to invert the logic of the endstop.

const bool Y\_MIN\_ENDSTOP\_INVERTING = true; // set to true to invert the logic of the endstop.

const bool Z\_MIN\_ENDSTOP\_INVERTING = false; // set to true to invert the logic of the endstop.

const bool X\_MAX\_ENDSTOP\_INVERTING = true; // set to true to invert the logic of the endstop.

const bool Y\_MAX\_ENDSTOP\_INVERTING = true; // set to true to invert the logic of the endstop.

const bool Z\_MAX\_ENDSTOP\_INVERTING = false; // set to true to invert the logic of the endstop.

//#define DISABLE\_MAX\_ENDSTOPS

//#define DISABLE\_MIN\_ENDSTOPS

// Disable max endstops for compatibility with endstop checking routine

#if defined(COREXY) && !defined(DISABLE\_MAX\_ENDSTOPS)

#define DISABLE\_MAX\_ENDSTOPS

#endif

// For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active High) use 1

#define X\_ENABLE\_ON 0

#define Y\_ENABLE\_ON 0

#define Z\_ENABLE\_ON 0

#define E\_ENABLE\_ON 0 // For all extruders

// Disables axis when it's not being used.

#define DISABLE\_X false

#define DISABLE\_Y false

#define DISABLE\_Z false

#define DISABLE\_E false // For all extruders

#define DISABLE\_INACTIVE\_EXTRUDER true //disable only inactive extruders and keep active extruder enabled

#define INVERT\_X\_DIR false // for Mendel set to false, for Orca set to true

#define INVERT\_Y\_DIR false // for Mendel set to true, for Orca set to false

#define INVERT\_Z\_DIR true // for Mendel set to false, for Orca set to true

#define INVERT\_E0\_DIR false // for direct drive extruder v9 set to true, for geared extruder set to false

#define INVERT\_E1\_DIR false // for direct drive extruder v9 set to true, for geared extruder set to false

#define INVERT\_E2\_DIR false // for direct drive extruder v9 set to true, for geared extruder set to false

// ENDSTOP SETTINGS:

// Sets direction of endstops when homing; 1=MAX, -1=MIN

#define X\_HOME\_DIR -1

#define Y\_HOME\_DIR -1

#define Z\_HOME\_DIR -1

#define min\_software\_endstops false // If true, axis won't move to coordinates less than HOME\_POS.

#define max\_software\_endstops true // If true, axis won't move to coordinates greater than the defined lengths below.

// Travel limits after homing

#define X\_MAX\_POS 220

#define X\_MIN\_POS 0

#define Y\_MAX\_POS 220

#define Y\_MIN\_POS 0

#define Z\_MAX\_POS 200

#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)

//============================= Bed Auto Leveling ===========================

// For TEVO Tarantula Auto-Level uncomment the 2 following lines not proved yet

#define ENABLE\_AUTO\_BED\_LEVELING // Delete the comment to enable (remove // at the start of the line)

#define Z\_PROBE\_REPEATABILITY\_TEST // If not commented out, Z-Probe Repeatability test will be included if Auto Bed Leveling is Enabled.

#ifdef ENABLE\_AUTO\_BED\_LEVELING

// There are 2 different ways to pick the X and Y locations to probe:

// - "grid" mode

// Probe every point in a rectangular grid

// You must specify the rectangle, and the density of sample points

// This mode is preferred because there are more measurements.

// It used to be called ACCURATE\_BED\_LEVELING but "grid" is more descriptive

// - "3-point" mode

// Probe 3 arbitrary points on the bed (that aren't colinear)

// You must specify the X & Y coordinates of all 3 points

#define AUTO\_BED\_LEVELING\_GRID

// with AUTO\_BED\_LEVELING\_GRID, the bed is sampled in a

// AUTO\_BED\_LEVELING\_GRID\_POINTSxAUTO\_BED\_LEVELING\_GRID\_POINTS grid

// and least squares solution is calculated

// Note: this feature occupies 10'206 byte

#ifdef AUTO\_BED\_LEVELING\_GRID

// set the rectangle in which to probe

#define LEFT\_PROBE\_BED\_POSITION 20

#define RIGHT\_PROBE\_BED\_POSITION 160

#define BACK\_PROBE\_BED\_POSITION 170

#define FRONT\_PROBE\_BED\_POSITION 20

// set the number of grid points per dimension

// I wouldn't see a reason to go above 3 (=9 probing points on the bed)

#define AUTO\_BED\_LEVELING\_GRID\_POINTS 2

#endif // AUTO\_BED\_LEVELING\_GRID

// these are the offsets to the probe relative to the extruder tip (Hotend - Probe)

#define X\_PROBE\_OFFSET\_FROM\_EXTRUDER -10

#define Y\_PROBE\_OFFSET\_FROM\_EXTRUDER -10

#define Z\_PROBE\_OFFSET\_FROM\_EXTRUDER -1.2

#define Z\_RAISE\_BEFORE\_HOMING 2 // (in mm) Raise Z before homing (G28) for Probe Clearance.

// Be sure you have this distance over your Z\_MAX\_POS in case

#define XY\_TRAVEL\_SPEED 8000 // X and Y axis travel speed between probes, in mm/min

#define Z\_RAISE\_BEFORE\_PROBING 2 //How much the extruder will be raised before traveling to the first probing point.

#define Z\_RAISE\_BETWEEN\_PROBINGS 2 //How much the extruder will be raised when traveling from between next probing points

//#define Z\_PROBE\_SLED // turn on if you have a z-probe mounted on a sled like those designed by Charles Bell

//#define SLED\_DOCKING\_OFFSET 5 // the extra distance the X axis must travel to pickup the sled. 0 should be fine but you can push it further if you'd like.

//If defined, the Probe servo will be turned on only during movement and then turned off to avoid jerk

//The value is the delay to turn the servo off after powered on - depends on the servo speed; 300ms is good value, but you can try lower it.

// You MUST HAVE the SERVO\_ENDSTOPS defined to use here a value higher than zero otherwise your code will not compile.

// #define PROBE\_SERVO\_DEACTIVATION\_DELAY 300

//If you have enabled the Bed Auto Leveling and are using the same Z Probe for Z Homing,

//it is highly recommended you let this Z\_SAFE\_HOMING enabled!!!

#define Z\_SAFE\_HOMING // This feature is meant to avoid Z homing with probe outside the bed area.

// When defined, it will:

// - Allow Z homing only after X and Y homing AND stepper drivers still enabled

// - If stepper drivers timeout, it will need X and Y homing again before Z homing

// - Position the probe in a defined XY point before Z Homing when homing all axis (G28)

// - Block Z homing only when the probe is outside bed area.

#ifdef Z\_SAFE\_HOMING

//#define Z\_SAFE\_HOMING\_X\_POINT -25

//#define Z\_SAFE\_HOMING\_Y\_POINT -25

#define Z\_SAFE\_HOMING\_X\_POINT (X\_MAX\_LENGTH/2) // X point for Z homing when homing all axis (G28)

#define Z\_SAFE\_HOMING\_Y\_POINT (Y\_MAX\_LENGTH/2) // Y point for Z homing when homing all axis (G28)

#endif

#ifdef AUTO\_BED\_LEVELING\_GRID // Check if Probe\_Offset \* Grid Points is greater than Probing Range

#if X\_PROBE\_OFFSET\_FROM\_EXTRUDER < 0

#if (-(X\_PROBE\_OFFSET\_FROM\_EXTRUDER \* AUTO\_BED\_LEVELING\_GRID\_POINTS) >= (RIGHT\_PROBE\_BED\_POSITION - LEFT\_PROBE\_BED\_POSITION))

#error "The X axis probing range is not enough to fit all the points defined in AUTO\_BED\_LEVELING\_GRID\_POINTS"

#endif

#else

#if ((X\_PROBE\_OFFSET\_FROM\_EXTRUDER \* AUTO\_BED\_LEVELING\_GRID\_POINTS) >= (RIGHT\_PROBE\_BED\_POSITION - LEFT\_PROBE\_BED\_POSITION))

#error "The X axis probing range is not enough to fit all the points defined in AUTO\_BED\_LEVELING\_GRID\_POINTS"

#endif

#endif

#if Y\_PROBE\_OFFSET\_FROM\_EXTRUDER < 0

#if (-(Y\_PROBE\_OFFSET\_FROM\_EXTRUDER \* AUTO\_BED\_LEVELING\_GRID\_POINTS) >= (BACK\_PROBE\_BED\_POSITION - FRONT\_PROBE\_BED\_POSITION))

#error "The Y axis probing range is not enough to fit all the points defined in AUTO\_BED\_LEVELING\_GRID\_POINTS"

#endif

#else

#if ((Y\_PROBE\_OFFSET\_FROM\_EXTRUDER \* AUTO\_BED\_LEVELING\_GRID\_POINTS) >= (BACK\_PROBE\_BED\_POSITION - FRONT\_PROBE\_BED\_POSITION))

#error "The Y axis probing range is not enough to fit all the points defined in AUTO\_BED\_LEVELING\_GRID\_POINTS"

#endif

#endif

#endif

#endif // ENABLE\_AUTO\_BED\_LEVELING

// The position of the homing switches

#define MANUAL\_HOME\_POSITIONS // If defined, MANUAL\_\*\_HOME\_POS below will be used

//#define BED\_CENTER\_AT\_0\_0 // If defined, the center of the bed is at (X=0, Y=0)

//Manual homing switch locations:

// For deltabots this means top and center of the Cartesian print volume.

// TEVO Tarantula Stock Settings - Home position

// #define MANUAL\_X\_HOME\_POS -11

// #define MANUAL\_Y\_HOME\_POS -41

// #define MANUAL\_Z\_HOME\_POS 0

// #define MANUAL\_Z\_HOME\_POS 402 // For delta: Distance between nozzle and print surface after homing.

// TEVO Tarantula Custom Settings - Home position - depend where you place endstops. Specially if big bed or not on Y.

#define MANUAL\_X\_HOME\_POS 0

#define MANUAL\_Y\_HOME\_POS -30

#define MANUAL\_Z\_HOME\_POS 0

//#define MANUAL\_Z\_HOME\_POS 402 // For delta: Distance between nozzle and print surface after homing.

//// MOVEMENT SETTINGS

#define NUM\_AXIS 4 // The axis order in all axis related arrays is X, Y, Z, E

#define HOMING\_FEEDRATE {50\*60, 50\*60, 4\*60, 0} // set the homing speeds (mm/min)

// default settings

// TEVO Tarantula Stock Settings - E-steps

//#define DEFAULT\_AXIS\_STEPS\_PER\_UNIT {80,80,1600,100} // default steps per unit for Tarantula //78.7402, 78.7402,1070,865.88

// TEVO Tarantula Custom Settings - E-steps

#define DEFAULT\_AXIS\_STEPS\_PER\_UNIT {80,80,1600,97.0874} // custom steps per unit for Tarantula

#define DEFAULT\_MAX\_FEEDRATE {225, 225, 3, 25} // (mm/sec)

// TEVO Tarantula Stock Settings - Acceleration

//#define DEFAULT\_MAX\_ACCELERATION {2650,2650,100,10000} // X, Y, Z, E maximum start speed for accelerated moves. E default values are good for skeinforge 40+, for older versions raise them a lot.

//#define DEFAULT\_ACCELERATION 2650 // X, Y, Z and E max acceleration in mm/s^2 for printing moves

// TEVO Tarantula Custom Settings - Acceleration

#define DEFAULT\_MAX\_ACCELERATION {1000,1000,100,10000} // X, Y, Z, E maximum start speed for accelerated moves. E default values are good for skeinforge 40+, for older versions raise them a lot.

#define DEFAULT\_ACCELERATION 1000 // X, Y, Z and E max acceleration in mm/s^2 for printing moves

#define DEFAULT\_RETRACT\_ACCELERATION 3000 // X, Y, Z and E max acceleration in mm/s^2 for retracts

// Offset of the extruders (uncomment if using more than one and relying on firmware to position when changing).

// The offset has to be X=0, Y=0 for the extruder 0 hotend (default extruder).

// For the other hotends it is their distance from the extruder 0 hotend.

// #define EXTRUDER\_OFFSET\_X {0.0, 20.00} // (in mm) for each extruder, offset of the hotend on the X axis

// #define EXTRUDER\_OFFSET\_Y {0.0, 5.00} // (in mm) for each extruder, offset of the hotend on the Y axis

// The speed change that does not require acceleration (i.e. the software might assume it can be done instantaneously)

#define DEFAULT\_XYJERK 15.0 // (mm/sec)

#define DEFAULT\_ZJERK 0.4 // (mm/sec)

#define DEFAULT\_EJERK 5.0 // (mm/sec)

//===========================================================================

//=============================Additional Features===========================

//===========================================================================

// Custom M code points

//#define CUSTOM\_M\_CODES

#ifdef CUSTOM\_M\_CODES

//#define CUSTOM\_M\_CODE\_SET\_Z\_PROBE\_OFFSET 851

//#define Z\_PROBE\_OFFSET\_RANGE\_MIN -15

//#define Z\_PROBE\_OFFSET\_RANGE\_MAX -5

#endif

// 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 210

#define PLA\_PREHEAT\_HPB\_TEMP 60

#define PLA\_PREHEAT\_FAN\_SPEED 0 // Insert Value between 0 and 255

#define ABS\_PREHEAT\_HOTEND\_TEMP 240

#define ABS\_PREHEAT\_HPB\_TEMP 100

#define ABS\_PREHEAT\_FAN\_SPEED 0 // 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 SD\_CHECK\_AND\_RETRY // Use CRC checks and retries on the SD communication

//#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\_MaKrP...

//#define MAKRPANEL

// The RepRapDiscount Smart Controller (white PCB)

// http://reprap.org/wiki/RepRapDiscou...

#define REPRAP\_DISCOUNT\_SMART\_CONTROLLER

// The GADGETS3D G3D LCD/SD Controller (blue PCB)

// http://reprap.org/wiki/RAMPS\_1.3/1....

//#define G3D\_PANEL

// The RepRapDiscount FULL GRAPHIC Smart Controller (quadratic white PCB)

// http://reprap.org/wiki/RepRapDiscou...

//

// ==> REMEMBER TO INSTALL U8glib to your ARDUINO library folder: http://code.google.com/p/u8glib/wik...

//#define REPRAP\_DISCOUNT\_FULL\_GRAPHIC\_SMART\_CONTROLLER

// The RepRapWorld REPRAPWORLD\_KEYPAD v1.1

// http://reprapworld.com/?products\_de...

//#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?rout...

// REMEMBER TO INSTALL LiquidCrystal\_I2C.h in your ARUDINO library folder: https://github.com/kiyoshigawa/Liqu...

//#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

#define ENCODER\_PULSES\_PER\_STEP 4

#define ENCODER\_STEPS\_PER\_MENU\_ITEM 1

#endif

#if defined(ULTIMAKERCONTROLLER) || defined(REPRAP\_DISCOUNT\_SMART\_CONTROLLER) || defined(G3D\_PANEL)

#define ULTIPANEL

#define NEWPANEL

#define ENCODER\_PULSES\_PER\_STEP 4

#define ENCODER\_STEPS\_PER\_MENU\_ITEM 1

#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

#ifdef LCD\_I2C\_SAINSMART\_YWROBOT

// This uses the LiquidCrystal\_I2C library ( https://bitbucket.org/fmalpartida/n... )

// 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

// PANELOLU2 LCD with status LEDs, separate encoder and click inputs

//#define LCD\_I2C\_PANELOLU2

#ifdef LCD\_I2C\_PANELOLU2

// This uses the LiquidTWI2 library v1.2.3 or later ( https://github.com/lincomatic/Liqui... )

// Make sure the LiquidTWI2 directory is placed in the Arduino or Sketchbook libraries subdirectory.

// (v1.2.3 no longer requires you to define PANELOLU in the LiquidTWI2.h library header file)

// Note: The PANELOLU2 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/Liqui... )

// 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/n...

//#define SAV\_3DLCD

#ifdef SAV\_3DLCD

#define SR\_LCD\_2W\_NL // Non latching 2 wire shiftregister

#define NEWPANEL

#define ULTIPANEL

#endif

#ifdef ULTIPANEL

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

#define SDSUPPORT

#define ULTRA\_LCD

#ifdef 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

#ifdef ULTRA\_LCD

#ifdef 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

#ifdef 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 bet 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\_h...

// #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 1 // 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, 10,90} // X,Y,Z Axis Extend and Retract angles

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

\* Support for a filament diameter sensor

\* Also allows adjustment of diameter at print time (vs at slicing)

\* Single extruder only at this point (extruder 0)

\*

\* Motherboards

\* 34 - RAMPS1.4 - uses Analog input 5 on the AUX2 connector

\* 81 - Printrboard - Uses Analog input 2 on the Exp1 connector (version B,C,D,E)

\* 301 - Rambo - uses Analog input 3

\* Note may require analog pins to be defined for different motherboards

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Uncomment below to enable

//#define FILAMENT\_SENSOR

#define FILAMENT\_SENSOR\_EXTRUDER\_NUM 0 //The number of the extruder that has the filament sensor (0,1,2)

#define MEASUREMENT\_DELAY\_CM 14 //measurement delay in cm. This is the distance from filament sensor to middle of barrel

#define DEFAULT\_NOMINAL\_FILAMENT\_DIA 1.75 //Enter the diameter (in mm) of the filament generally used (3.0 mm or 1.75 mm) - this is then used in the slicer software. Used for sensor reading validation

#define MEASURED\_UPPER\_LIMIT 2 //upper limit factor used for sensor reading validation in mm

#define MEASURED\_LOWER\_LIMIT 1 //lower limit factor for sensor reading validation in mm

#define MAX\_MEASUREMENT\_DELAY 20 //delay buffer size in bytes (1 byte = 1cm)- limits maximum measurement delay allowable (must be larger than MEASUREMENT\_DELAY\_CM and lower number saves RAM)

//defines used in the code

#define DEFAULT\_MEASURED\_FILAMENT\_DIA DEFAULT\_NOMINAL\_FILAMENT\_DIA //set measured to nominal initially

//When using an LCD, uncomment the line below to display the Filament sensor data on the last line instead of status. Status will appear for 5 sec.

//#define FILAMENT\_LCD\_DISPLAY

#include "Configuration\_adv.h"

#include "thermistortables.h"

#endif //\_\_CONFIGURATION\_H