```
/**
 * Marlin 3D Printer Firmware
 * Copyright (C) 2016 MarlinFirmware
[https://github.com/MarlinFirmware/Marlin]
 *
 * Based on Sprinter and grbl.
 * Copyright (C) 2011 Camiel Gubbels / Erik van der Zalm
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program.  If not, see <http://www.gnu.org/licenses/>.
 *
 */

/**
 * Configuration.h
 *
 * Basic settings such as:
 *
 * - Type of electronics
 * - Type of temperature sensor
 * - Printer geometry
 * - Endstop configuration
 * - LCD controller
 * - Extra features
 *
 * Advanced settings can be found in Configuration_adv.h
 *
 */
#ifndef CONFIGURATION_H
#define CONFIGURATION_H
#define CONFIGURATION_H_VERSION 010107

//=========================================================================
====
//============================= Getting Started
============================
//=========================================================================
====

/**
 * Here are some standard links for getting your machine calibrated:
 *
 * http://reprap.org/wiki/Calibration
 * http://youtu.be/wAL9d7FgInk
 * http://calculator.josefprusa.cz
 * http://reprap.org/wiki/Triffid_Hunter%27s_Calibration_Guide
 * http://www.thingiverse.com/thing:5573
 * https://sites.google.com/site/repraplogphase/calibration-of-your-
reprap
 * http://www.thingiverse.com/thing:298812
 */

//===========================================================================
====
//============================= DELTA Printer
==============================
//===========================================================================
====
// For a Delta printer start with one of the configuration files in the
// example_configurations/delta directory and customize for your machine.
//

//===========================================================================
====
//============================= SCARA Printer
==============================
//===========================================================================
====
// For a SCARA printer start with the configuration files in
// example_configurations/SCARA and customize for your machine.
//

// @section info

// 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_CONFIG_H_AUTHOR "(none, default config)" // Who made the
changes.
#define SHOW_BOOTSCREEN
#define STRING_SPLASH_LINE1 SHORT_BUILD_VERSION // will be shown during
bootup in line 1
#define STRING_SPLASH_LINE2 WEBSITE_URL         // will be shown during
bootup in line 2

//
// *** VENDORS PLEASE READ
*****************************************************
//
// Marlin now allow you to have a vendor boot image to be displayed on
machine
// start. When SHOW_CUSTOM_BOOTSCREEN is defined Marlin will first show
your
// custom boot image and then the default Marlin boot image is shown.
//
// We suggest for you to take advantage of this new feature and keep the
Marlin
// boot image unmodified. For an example have a look at the bq Hephestos
2
// example configuration folder.
//
//#define SHOW_CUSTOM_BOOTSCREEN
// @section machine

/**
 * Select which serial port on the board will 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 always used by the Arduino bootloader regardless of
this setting.
 *
 * :[0, 1, 2, 3, 4, 5, 6, 7]
 */
#define SERIAL_PORT 0

/**
 * This setting determines the communication speed of the printer.
 *
 * 250000 works in most cases, but you might try a lower speed if
 * you commonly experience drop-outs during host printing.
 * You may try up to 1000000 to speed up SD file transfer.
 *
 * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
 */
#define BAUDRATE 250000

// Enable the Bluetooth serial interface on AT90USB devices
//#define BLUETOOTH

// The following define selects which electronics board you have.
// Please choose the name from boards.h that matches your setup
#ifndef MOTHERBOARD
  #define MOTHERBOARD BOARD_RAMPS_14_EFB
#endif

// Optional custom name for your RepStrap or other custom machine
// Displayed in the LCD "Ready" message
#define CUSTOM_MACHINE_NAME "Naktrem Deltabot"

// 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/version4)
//#define MACHINE_UUID "00000000-0000-0000-0000-000000000000"

// @section extruder

// This defines the number of extruders
// :[1, 2, 3, 4, 5]
#define EXTRUDERS 1

// Generally expected filament diameter (1.75, 2.85, 3.0, ...). Used for
Volumetric, Filament Width Sensor, etc.
#define DEFAULT_NOMINAL_FILAMENT_DIA 3.0

// For Cyclops or any "multi-extruder" that shares a single nozzle.
//#define SINGLENOZZLE

/**
 * PrÅ¯Å¡a MK2 Single Nozzle Multi-Material Multiplexer, and variants.
 *
 * This device allows one stepper driver on a control board to drive
 * two to eight stepper motors, one at a time, in a manner suitable
 * for extruders.
 *
 * This option only allows the multiplexer to switch on tool-change.
 * Additional options to configure custom E moves are pending.
 */
//#define MK2_MULTIPLEXER
#if ENABLED(MK2_MULTIPLEXER)
  // Override the default DIO selector pins here, if needed.
  // Some pins files may provide defaults for these pins.
  //#define E_MUX0_PIN 40  // Always Required
  //#define E_MUX1_PIN 42  // Needed for 3 to 8 steppers
  //#define E_MUX2_PIN 44  // Needed for 5 to 8 steppers
#endif

// A dual extruder that uses a single stepper motor
//#define SWITCHING_EXTRUDER
#if ENABLED(SWITCHING_EXTRUDER)
  #define SWITCHING_EXTRUDER_SERVO_NR 0
  #define SWITCHING_EXTRUDER_SERVO_ANGLES { 0, 90 } // Angles for E0,
E1[, E2, E3]
  #if EXTRUDERS > 3
    #define SWITCHING_EXTRUDER_E23_SERVO_NR 1
  #endif
#endif

// A dual-nozzle that uses a servomotor to raise/lower one of the nozzles
//#define SWITCHING_NOZZLE
#if ENABLED(SWITCHING_NOZZLE)
  #define SWITCHING_NOZZLE_SERVO_NR 0
  #define SWITCHING_NOZZLE_SERVO_ANGLES { 0, 90 }   // Angles for E0, E1
  //#define HOTEND_OFFSET_Z { 0.0, 0.0 }
#endif

/**
 * Two separate X-carriages with extruders that connect to a moving part
 * via a magnetic docking mechanism. Requires SOL1_PIN and SOL2_PIN.
 */
//#define PARKING_EXTRUDER
#if ENABLED(PARKING_EXTRUDER)
  #define PARKING_EXTRUDER_SOLENOIDS_INVERT           // If enabled, the
solenoid is NOT magnetized with applied voltage
  #define PARKING_EXTRUDER_SOLENOIDS_PINS_ACTIVE LOW  // LOW or HIGH pin
signal energizes the coil
  #define PARKING_EXTRUDER_SOLENOIDS_DELAY 250        // Delay (ms) for
magnetic field. No delay if 0 or not defined.
  #define PARKING_EXTRUDER_PARKING_X { -78, 184 }     // X positions for
parking the extruders
  #define PARKING_EXTRUDER_GRAB_DISTANCE 1            // mm to move
beyond the parking point to grab the extruder
  #define PARKING_EXTRUDER_SECURITY_RAISE 5           // Z-raise before
parking
  #define HOTEND_OFFSET_Z { 0.0, 1.3 }                // Z-offsets of the
two hotends. The first must be 0.
#endif

/**
 * "Mixing Extruder"
 *   - Adds a new code, M165, to set the current mix factors.
 *   - Extends the stepping routines to move multiple steppers in
proportion to the mix.
 *   - Optional support for Repetier Firmware M163, M164, and virtual
extruder.
 *   - This implementation supports only a single extruder.
 *   - Enable DIRECT_MIXING_IN_G1 for Pia Taubert's reference
implementation
 */
//#define MIXING_EXTRUDER
#if ENABLED(MIXING_EXTRUDER)
  #define MIXING_STEPPERS 2        // Number of steppers in your mixing
extruder
  #define MIXING_VIRTUAL_TOOLS 16  // Use the Virtual Tool method with
M163 and M164
  //#define DIRECT_MIXING_IN_G1    // Allow ABCDHI mix factors in G1
movement commands
#endif

// 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 HOTEND_OFFSET_X {0.0, 20.00} // (in mm) for each extruder,
offset of the hotend on the X axis
//#define HOTEND_OFFSET_Y {0.0, 5.00}  // (in mm) for each extruder,
offset of the hotend on the Y axis

// @section machine

/**
 * Select your power supply here. Use 0 if you haven't connected the
PS_ON_PIN
 *
 * 0 = No Power Switch
 * 1 = ATX
 * 2 = X-Box 360 203Watts (the blue wire connected to PS_ON and the red
wire to VCC)
 *
 * :{ 0:'No power switch', 1:'ATX', 2:'X-Box 360' }
 */
#define POWER_SUPPLY 1

#if POWER_SUPPLY > 0
  // Enable this option to leave the PSU off at startup.
  // Power to steppers and heaters will need to be turned on with M80.
  //#define PS_DEFAULT_OFF
#endif

// @section temperature

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

/**
 * --NORMAL IS 4.7kohm PULLUP!-- 1kohm pullup can be used on hotend
sensor, using correct resistor and table
 *
 * Temperature sensors available:
 *
 *    -3 : thermocouple with MAX31855 (only for sensor 0)
 *    -2 : thermocouple with MAX6675 (only for sensor 0)
 *    -1 : thermocouple with AD595
 *     0 : not used
 *     1 : 100k thermistor - best choice for EPCOS 100k (4.7k pullup)
 *     2 : 200k thermistor - ATC Semitec 204GT-2 (4.7k pullup)
 *     3 : Mendel-parts thermistor (4.7k pullup)
 *     4 : 10k thermistor !! do not use it for a hotend. It gives bad
resolution at high temp. !!
 *     5 : 100K thermistor - ATC Semitec 104GT-2 (Used in ParCan & J-
Head) (4.7k pullup)
 *     6 : 100k EPCOS - Not as accurate as table 1 (created using a fluke
thermocouple) (4.7k pullup)
 *     7 : 100k Honeywell thermistor 135-104LAG-J01 (4.7k pullup)
 *    71 : 100k Honeywell thermistor 135-104LAF-J01 (4.7k pullup)
 *     8 : 100k 0603 SMD Vishay NTCS0603E3104FXT (4.7k pullup)
 *     9 : 100k GE Sensing AL03006-58.2K-97-G1 (4.7k pullup)
 *    10 : 100k RS thermistor 198-961 (4.7k pullup)
 *    11 : 100k beta 3950 1% thermistor (4.7k pullup)
 *    12 : 100k 0603 SMD Vishay NTCS0603E3104FXT (4.7k pullup)
(calibrated for Makibox hot bed)
 *    13 : 100k Hisens 3950  1% up to 300Â°C for hotend "Simple ONE " &
"Hotend "All In ONE"
 *    20 : the PT100 circuit found in the Ultimainboard V2.x
 *    60 : 100k Maker's Tool Works Kapton Bed Thermistor beta=3950
 *    66 : 4.7M High Temperature thermistor from Dyze Design
 *    70 : the 100K thermistor found in the bq Hephestos 2
 *    75 : 100k Generic Silicon Heat Pad with NTC 100K MGB18-104F39050L32
thermistor
 *
 *       1k ohm pullup tables - This is atypical, and requires changing
out the 4.7k pullup for 1k.
 *                             (but gives greater accuracy and more
stable PID)
 *    51 : 100k thermistor - EPCOS (1k pullup)
 *    52 : 200k thermistor - ATC Semitec 204GT-2 (1k pullup)
 *    55 : 100k thermistor - ATC Semitec 104GT-2 (Used in ParCan & J-
Head) (1k pullup)
 *
 *  1047 : Pt1000 with 4k7 pullup
 *  1010 : Pt1000 with 1k pullup (non standard)
 *   147 : Pt100 with 4k7 pullup
 *   110 : Pt100 with 1k pullup (non standard)
 *
 *         Use these for Testing or Development purposes. NEVER for
production machine.
 *   998 : Dummy Table that ALWAYS reads 25Â°C or the temperature defined
below.
 *   999 : Dummy Table that ALWAYS reads 100Â°C or the temperature
defined below.
 *
 * :{ '0': "Not used", '1':"100k / 4.7k - EPCOS", '2':"200k / 4.7k - ATC
Semitec 204GT-2", '3':"Mendel-parts / 4.7k", '4':"10k !! do not use for a
hotend. Bad resolution at high temp. !!", '5':"100K / 4.7k - ATC Semitec
104GT-2 (Used in ParCan & J-Head)", '6':"100k / 4.7k EPCOS - Not as
accurate as Table 1", '7':"100k / 4.7k Honeywell 135-104LAG-J01",
'8':"100k / 4.7k 0603 SMD Vishay NTCS0603E3104FXT", '9':"100k / 4.7k GE
Sensing AL03006-58.2K-97-G1", '10':"100k / 4.7k RS 198-961", '11':"100k /
4.7k beta 3950 1%", '12':"100k / 4.7k 0603 SMD Vishay NTCS0603E3104FXT
(calibrated for Makibox hot bed)", '13':"100k Hisens 3950  1% up to
300Â°C for hotend 'Simple ONE ' & hotend 'All In ONE'", '20':"PT100
(Ultimainboard V2.x)", '51':"100k / 1k - EPCOS", '52':"200k / 1k - ATC
Semitec 204GT-2", '55':"100k / 1k - ATC Semitec 104GT-2 (Used in ParCan &
J-Head)", '60':"100k Maker's Tool Works Kapton Bed Thermistor beta=3950",
'66':"Dyze Design 4.7M High Temperature thermistor", '70':"the 100K
thermistor found in the bq Hephestos 2", '71':"100k / 4.7k Honeywell 135-
104LAF-J01", '147':"Pt100 / 4.7k", '1047':"Pt1000 / 4.7k", '110':"Pt100 /
1k (non-standard)", '1010':"Pt1000 / 1k (non standard)", '-
3':"Thermocouple + MAX31855 (only for sensor 0)", '-2':"Thermocouple +
MAX6675 (only for sensor 0)", '-1':"Thermocouple + AD595",'998':"Dummy
1", '999':"Dummy 2" }
 */
#define TEMP_SENSOR_0 5
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_3 0
#define TEMP_SENSOR_4 0
#define TEMP_SENSOR_BED 0

// Dummy thermistor constant temperature readings, for use with 998 and
999
#define DUMMY_THERMISTOR_998_VALUE 25
#define DUMMY_THERMISTOR_999_VALUE 100

// Use temp sensor 1 as a redundant sensor with sensor 0. If the readings
// from the two sensors differ too much the print will be aborted.
//#define TEMP_SENSOR_1_AS_REDUNDANT
#define MAX_REDUNDANT_TEMP_SENSOR_DIFF 10

// Extruder temperature must be close to target for this long before M109
returns success
#define TEMP_RESIDENCY_TIME 10  // (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.

// Bed temperature must be close to target for this long before M190
returns success
#define TEMP_BED_RESIDENCY_TIME 0   // (seconds)
#define TEMP_BED_HYSTERESIS 3       // (degC) range of +/- temperatures
considered "close" to the target one
#define TEMP_BED_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 HEATER_3_MINTEMP 5
#define HEATER_4_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 275
```

```
#define HEATER_1_MAXTEMP 275
#define HEATER_2_MAXTEMP 275
#define HEATER_3_MAXTEMP 275
#define HEATER_4_MAXTEMP 275
#define BED_MAXTEMP 150

//==========================================================================
====
//============================ PID Settings
================================
//==========================================================================
====
// PID Tuning Guide here: http://reprap.org/wiki/PID_Tuning

// 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
#define PID_K1 0.95      // Smoothing factor within the PID
#if ENABLED(PIDTEMP)
  #define PID_AUTOTUNE_MENU // Add PID Autotune to the LCD "Temperature"
menu to run M303 and apply the result.
  //#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 SLOW_PWM_HEATERS // PWM with very low frequency (roughly
0.125Hz=8s) and minimum state time of approximately 1s useful for heaters
driven by a relay
  //#define PID_PARAMS_PER_HOTEND // Uses separate PID parameters for
each extruder (useful for mismatched extruders)
                                  // Set/get with gcode: M301 E[extruder
number, 0-2]
  #define PID_FUNCTIONAL_RANGE 10 // If the temperature difference
between the target temperature and the actual temperature
                                  // is more than PID_FUNCTIONAL_RANGE
then the PID will be shut off and the heater will be set to min/max.

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

  // Ultimaker
  #define  DEFAULT_Kp 22.2
  #define  DEFAULT_Ki 1.08
  #define  DEFAULT_Kd 114

  // 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
```

```
//=============================================================================
====
//============================= PID > 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 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

#if ENABLED(PIDTEMPBED)

  //#define PID_BED_DEBUG // Sends debug data to the serial port.

  //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)
  #define  DEFAULT_bedKp 10.00
  #define  DEFAULT_bedKi .023
  #define  DEFAULT_bedKd 305.4

  //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

// @section extruder

// This option prevents extrusion if the temperature is below
EXTRUDE_MINTEMP.
```

```
// It also enables the M302 command to set the minimum extrusion
temperature
// or to allow moving the extruder regardless of the hotend temperature.
// *** IT IS HIGHLY RECOMMENDED TO LEAVE THIS OPTION ENABLED! ***
#define PREVENT_COLD_EXTRUSION
#define EXTRUDE_MINTEMP 170

// This option prevents a single extrusion longer than EXTRUDE_MAXLENGTH.
// Note that for Bowden Extruders a too-small value here may prevent
loading.
#define PREVENT_LENGTHY_EXTRUDE
#define EXTRUDE_MAXLENGTH 200

//===========================================================================
====
//======================= Thermal Runaway Protection
======================
//===========================================================================
====

/**
 * Thermal Protection provides additional protection to your printer from
damage
 * and fire. Marlin always includes safe min and max temperature ranges
which
 * protect against a broken or disconnected thermistor wire.
 *
 * The issue: If a thermistor falls out, it will report the much lower
 * temperature of the air in the room, and the the firmware will keep
 * the heater on.
 *
 * If you get "Thermal Runaway" or "Heating failed" errors the
 * details can be tuned in Configuration_adv.h
 */

#define THERMAL_PROTECTION_HOTENDS // Enable thermal protection for all
extruders
#define THERMAL_PROTECTION_BED     // Enable thermal protection for the
heated bed

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

// @section machine

// Uncomment one of these options to enable CoreXY, CoreXZ, or CoreYZ
kinematics
// either in the usual order or reversed
//#define COREXY
//#define COREXZ
//#define COREYZ
//#define COREYX
//#define COREZX
//#define COREZY
```

```
//===========================================================================
====
//============================== Delta Settings
=============================
//===========================================================================
====
// Enable DELTA kinematics and most of the default configuration for
Deltas
#define DELTA

#if ENABLED(DELTA)

  // Make delta curves from many straight lines (linear interpolation).
  // This is a trade-off between visible corners (not enough segments)
  // and processor overload (too many expensive sqrt calls).
  #define DELTA_SEGMENTS_PER_SECOND 2160

  // After homing move down to a height where XY movement is
unconstrained
  //#define DELTA_HOME_TO_SAFE_ZONE

  // Delta calibration menu
  // uncomment to add three points calibration menu option.
  // See http://minow.blogspot.com/index.html#4918805519571907051
  #define DELTA_CALIBRATION_MENU

  // uncomment to add G33 Delta Auto-Calibration (Enable EEPROM_SETTINGS
to store results)
  #define DELTA_AUTO_CALIBRATION

  // NOTE NB all values for DELTA_* values MUST be floating point, so
always have a decimal point in them

  #if ENABLED(DELTA_AUTO_CALIBRATION)
    // set the default number of probe points : n*n (1 -> 7)
    #define DELTA_CALIBRATION_DEFAULT_POINTS 4

    // Enable and set these values based on results of 'G33 A'
    //#define H_FACTOR 1.01
    //#define R_FACTOR 2.61
    //#define A_FACTOR 0.87

  #endif

  #if ENABLED(DELTA_AUTO_CALIBRATION) || ENABLED(DELTA_CALIBRATION_MENU)
    // Set the radius for the calibration probe points - max 0.9 *
DELTA_PRINTABLE_RADIUS for non-eccentric probes
    #define DELTA_CALIBRATION_RADIUS 85.0 // mm
    // Set the steprate for papertest probing
    #define PROBE_MANUALLY_STEP 0.025
  #endif

  // Print surface diameter/2 minus unreachable space (avoid collisions
with vertical towers).
  #define DELTA_PRINTABLE_RADIUS 85.0 // mm

  // Center-to-center distance of the holes in the diagonal push rods.
  #define DELTA_DIAGONAL_ROD 330.0 // mm

  // height from z=0 to home position
  #define DELTA_HEIGHT 295.00 // get this value from auto calibrate

  #define DELTA_ENDSTOP_ADJ { 0.0, 0.0, 0.0 } // get these from auto
calibrate

  // Horizontal distance bridged by diagonal push rods when effector is
centered.
  #define DELTA_RADIUS 85.0 //mm  Get this value from auto calibrate

  // Trim adjustments for individual towers
  // tower angle corrections for X and Y tower / rotate XYZ so Z tower
angle = 0
  // measured in degrees anticlockwise looking from above the printer
  #define DELTA_TOWER_ANGLE_TRIM { 0.0, 0.0, 0.0 } // get these values
from auto calibrate

  // delta radius and diaginal rod adjustments measured in mm
  //#define DELTA_RADIUS_TRIM_TOWER { 0.0, 0.0, 0.0 }
  //#define DELTA_DIAGONAL_ROD_TRIM_TOWER { 0.0, 0.0, 0.0 }

#endif

//=========================================================================
====
//============================== Endstop Settings
=======================
//=========================================================================
====

// @section homing

// Specify here all the endstop connectors that are connected to any
endstop or probe.
// Almost all printers will be using one per axis. Probes will use one or
more of the
// extra connectors. Leave undefined any used for non-endstop and non-
probe purposes.
//#define USE_XMIN_PLUG
//#define USE_YMIN_PLUG
#define USE_ZMIN_PLUG // a Z probe
#define USE_XMAX_PLUG
#define USE_YMAX_PLUG
#define USE_ZMAX_PLUG

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

#if DISABLED(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
  //#define ENDSTOPPULLUP_ZMIN_PROBE
#endif
```

```
// Mechanical endstop with COM to ground and NC to Signal uses "false"
here (most common setup).
#define X_MIN_ENDSTOP_INVERTING true  // set to true to invert the logic
of the endstop.
#define Y_MIN_ENDSTOP_INVERTING true  // set to true to invert the logic
of the endstop.
#define Z_MIN_ENDSTOP_INVERTING true  // set to true to invert the logic
of the endstop.
#define X_MAX_ENDSTOP_INVERTING true  // set to true to invert the logic
of the endstop.
#define Y_MAX_ENDSTOP_INVERTING true  // set to true to invert the logic
of the endstop.
#define Z_MAX_ENDSTOP_INVERTING true  // set to true to invert the logic
of the endstop.
#define Z_MIN_PROBE_ENDSTOP_INVERTING true  // set to true to invert the
logic of the probe.

// Enable this feature if all enabled endstop pins are interrupt-capable.
// This will remove the need to poll the interrupt pins, saving many CPU
cycles.
//#define ENDSTOP_INTERRUPTS_FEATURE

//==========================================================================
======
//============================== Movement Settings
============================
//==========================================================================
======
// @section motion

// delta speeds must be the same on xyz
/**
 * Default Settings
 *
 * These settings can be reset by M502
 *
 * Note that if EEPROM is enabled, saved values will override these.
 */

/**
 * With this option each E stepper can have its own factors for the
 * following movement settings. If fewer factors are given than the
 * total number of extruders, the last value applies to the rest.
 */
//#define DISTINCT_E_FACTORS

/**
 * Default Axis Steps Per Unit (steps/mm)
 * Override with M92
 *                                      X, Y, Z, E0 [, E1[, E2[, E3[,
E4]]]]
 */
#define DEFAULT_AXIS_STEPS_PER_UNIT   { 160, 160, 160, 760*1.1 }  //
default steps per unit for Kossel (GT2, 20 tooth)

/**
 * Default Max Feed Rate (mm/s)
 * Override with M203
 *                                      X, Y, Z, E0 [, E1[, E2[, E3[,
E4]]]]
```

```
 */
#define DEFAULT_MAX_FEEDRATE          { 500, 500, 500, 25 }

/**
 * Default Max Acceleration (change/s) change = mm/s
 * (Maximum start speed for accelerated moves)
 * Override with M201
 *                                      X, Y, Z, E0 [, E1[, E2[, E3[,
E4]]]]
 */
#define DEFAULT_MAX_ACCELERATION      { 9000, 9000, 9000, 10000 }

/**
 * Default Acceleration (change/s) change = mm/s
 * Override with M204
 *
 *   M204 P    Acceleration
 *   M204 R    Retract Acceleration
 *   M204 T    Travel Acceleration
 */
#define DEFAULT_ACCELERATION          1000    // X, Y, Z and E
acceleration for printing moves
#define DEFAULT_RETRACT_ACCELERATION  3000    // E acceleration for
retracts
#define DEFAULT_TRAVEL_ACCELERATION   2000    // X, Y, Z acceleration for
travel (non printing) moves

/**
 * Default Jerk (mm/s)
 * Override with M205 X Y Z E
 *
 * "Jerk" specifies the minimum speed change that requires acceleration.
 * When changing speed and direction, if the difference is less than the
 * value set here, it may happen instantaneously.
 */
#define DEFAULT_XJERK                 10.0
#define DEFAULT_YJERK                 DEFAULT_XJERK
#define DEFAULT_ZJERK                 DEFAULT_XJERK // Must be same as XY
for delta
#define DEFAULT_EJERK                  5.0

//=========================================================================
====
//============================= Z Probe Options
============================
//=========================================================================
====
// @section probes

//
// See http://marlinfw.org/docs/configuration/probes.html
//

/**
 * Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
 *
 * Enable this option for a probe connected to the Z Min endstop pin.
 */
#define Z_MIN_PROBE_USES_Z_MIN_ENDSTOP_PIN
```

```
/**
 * Z_MIN_PROBE_ENDSTOP
 *
 * Enable this option for a probe connected to any pin except Z-Min.
 * (By default Marlin assumes the Z-Max endstop pin.)
 * To use a custom Z Probe pin, set Z_MIN_PROBE_PIN below.
 *
 *  - The simplest option is to use a free endstop connector.
 *  - Use 5V for powered (usually inductive) sensors.
 *
 *  - RAMPS 1.3/1.4 boards may use the 5V, GND, and Aux4->D32 pin:
 *    - For simple switches connect...
 *      - normally-closed switches to GND and D32.
 *      - normally-open switches to 5V and D32.
 *
 * WARNING: Setting the wrong pin may have unexpected and potentially
 * disastrous consequences. Use with caution and do your homework.
 *
 */
//#define Z_MIN_PROBE_ENDSTOP

/**
 * Probe Type
 *
 * Allen Key Probes, Servo Probes, Z-Sled Probes, FIX_MOUNTED_PROBE, etc.
 * Activate one of these to use Auto Bed Leveling below.
 */

/**
 * The "Manual Probe" provides a means to do "Auto" Bed Leveling without
a probe.
 * Use G29 repeatedly, adjusting the Z height at each point with movement
commands
 * or (with LCD_BED_LEVELING) the LCD controller.
 */
#define PROBE_MANUALLY

/**
 * A Fix-Mounted Probe either doesn't deploy or needs manual deployment.
 *   (e.g., an inductive probe or a nozzle-based probe-switch.)
 */
//#define FIX_MOUNTED_PROBE

/**
 * Z Servo Probe, such as an endstop switch on a rotating arm.
 */
//#define Z_ENDSTOP_SERVO_NR 0   // Defaults to SERVO 0 connector.
//#define Z_SERVO_ANGLES {70,0}  // Z Servo Deploy and Stow angles

/**
 * The BLTouch probe uses a Hall effect sensor and emulates a servo.
 */
//#define BLTOUCH
#if ENABLED(BLTOUCH)
  //#define BLTOUCH_DELAY 375   // (ms) Enable and increase if needed
#endif

/**
 * Enable one or more of the following if probing seems unreliable.
 * Heaters and/or fans can be disabled during probing to minimize
electrical
 * noise. A delay can also be added to allow noise and vibration to
settle.
 * These options are most useful for the BLTouch probe, but may also
improve
 * readings with inductive probes and piezo sensors.
 */
//#define PROBING_HEATERS_OFF       // Turn heaters off when probing
//#define PROBING_FANS_OFF          // Turn fans off when probing
//#define DELAY_BEFORE_PROBING 200  // (ms) To prevent vibrations from
triggering piezo sensors

// A probe that is deployed and stowed with a solenoid pin (SOL1_PIN)
//#define SOLENOID_PROBE

// A sled-mounted probe like those designed by Charles Bell.
//#define Z_PROBE_SLED
//#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.

//
// For Z_PROBE_ALLEN_KEY see the Delta example configurations.
//

/**
 *   Z Probe to nozzle (X,Y) offset, relative to (0, 0).
 *   X and Y offsets must be integers.
 *
 *   In the following example the X and Y offsets are both positive:
 *   #define X_PROBE_OFFSET_FROM_EXTRUDER 10
 *   #define Y_PROBE_OFFSET_FROM_EXTRUDER 10
 *
 *      +-- BACK ---+
 *      |           |
 *    L |    (+) P  | R <-- probe (20,20)
 *    E |           | I
 *    F | (-) N (+) | G <-- nozzle (10,10)
 *    T |           | H
 *      |    (-)    | T
 *      |           |
 *      O-- FRONT --+
 *    (0,0)
 */
#define X_PROBE_OFFSET_FROM_EXTRUDER 0     // X offset: -left  +right
[of the nozzle]
#define Y_PROBE_OFFSET_FROM_EXTRUDER -10   // Y offset: -front +behind
[the nozzle]
#define Z_PROBE_OFFSET_FROM_EXTRUDER -3.5  // Z offset: -below +above
[the nozzle]

// X and Y axis travel speed (mm/m) between probes
#define XY_PROBE_SPEED 4000

// Speed for the first approach when double-probing (MULTIPLE_PROBING ==
2)
#define Z_PROBE_SPEED_FAST HOMING_FEEDRATE_Z

// Speed for the "accurate" probe of each point
```

```
#define Z_PROBE_SPEED_SLOW (Z_PROBE_SPEED_FAST / 2)

// The number of probes to perform at each point.
//   Set to 2 for a fast/slow probe, using the second probe result.
//   Set to 3 or more for slow probes, averaging the results.
//#define MULTIPLE_PROBING 2

/**
 * Allen key retractable z-probe as seen on many Kossel delta printers -
http://reprap.org/wiki/Kossel#Automatic_bed_leveling_probe
 * Deploys by touching z-axis belt. Retracts by pushing the probe down.
Uses Z_MIN_PIN.
 */
//#define Z_PROBE_ALLEN_KEY

#if ENABLED(Z_PROBE_ALLEN_KEY)
  // 2 or 3 sets of coordinates for deploying and retracting the spring
loaded touch probe on G29,
  // if servo actuated touch probe is not defined. Uncomment as
appropriate for your printer/probe.

  #define Z_PROBE_ALLEN_KEY_DEPLOY_1_X 30.0
  #define Z_PROBE_ALLEN_KEY_DEPLOY_1_Y DELTA_PRINTABLE_RADIUS
  #define Z_PROBE_ALLEN_KEY_DEPLOY_1_Z 100.0
  #define Z_PROBE_ALLEN_KEY_DEPLOY_1_FEEDRATE XY_PROBE_SPEED

  #define Z_PROBE_ALLEN_KEY_DEPLOY_2_X 0.0
  #define Z_PROBE_ALLEN_KEY_DEPLOY_2_Y DELTA_PRINTABLE_RADIUS
  #define Z_PROBE_ALLEN_KEY_DEPLOY_2_Z 100.0
  #define Z_PROBE_ALLEN_KEY_DEPLOY_2_FEEDRATE (XY_PROBE_SPEED)/10

  #define Z_PROBE_ALLEN_KEY_DEPLOY_3_X Z_PROBE_ALLEN_KEY_DEPLOY_2_X *
0.75
  #define Z_PROBE_ALLEN_KEY_DEPLOY_3_Y Z_PROBE_ALLEN_KEY_DEPLOY_2_Y *
0.75
  #define Z_PROBE_ALLEN_KEY_DEPLOY_3_Z Z_PROBE_ALLEN_KEY_DEPLOY_2_Z
  #define Z_PROBE_ALLEN_KEY_DEPLOY_3_FEEDRATE XY_PROBE_SPEED

  #define Z_PROBE_ALLEN_KEY_STOW_1_X -64.0 // Move the probe into
position
  #define Z_PROBE_ALLEN_KEY_STOW_1_Y 56.0
  #define Z_PROBE_ALLEN_KEY_STOW_1_Z 23.0
  #define Z_PROBE_ALLEN_KEY_STOW_1_FEEDRATE XY_PROBE_SPEED

  #define Z_PROBE_ALLEN_KEY_STOW_2_X -64.0 // Push it down
  #define Z_PROBE_ALLEN_KEY_STOW_2_Y 56.0
  #define Z_PROBE_ALLEN_KEY_STOW_2_Z 3.0
  #define Z_PROBE_ALLEN_KEY_STOW_2_FEEDRATE (XY_PROBE_SPEED)/10

  #define Z_PROBE_ALLEN_KEY_STOW_3_X -64.0 // Move it up to clear
  #define Z_PROBE_ALLEN_KEY_STOW_3_Y 56.0
  #define Z_PROBE_ALLEN_KEY_STOW_3_Z 50.0
  #define Z_PROBE_ALLEN_KEY_STOW_3_FEEDRATE XY_PROBE_SPEED

  #define Z_PROBE_ALLEN_KEY_STOW_4_X 0.0
  #define Z_PROBE_ALLEN_KEY_STOW_4_Y 0.0
  #define Z_PROBE_ALLEN_KEY_STOW_4_Z Z_PROBE_ALLEN_KEY_STOW_3_Z
  #define Z_PROBE_ALLEN_KEY_STOW_4_FEEDRATE XY_PROBE_SPEED

#endif // Z_PROBE_ALLEN_KEY
```

```
/**
 * Z probes require clearance when deploying, stowing, and moving between
 * probe points to avoid hitting the bed and other hardware.
 * Servo-mounted probes require extra space for the arm to rotate.
 * Inductive probes need space to keep from triggering early.
 *
 * Use these settings to specify the distance (mm) to raise the probe (or
 * lower the bed). The values set here apply over and above any
(negative)
 * probe Z Offset set with Z_PROBE_OFFSET_FROM_EXTRUDER, M851, or the
LCD.
 * Only integer values >= 1 are valid here.
 *
 * Example: `M851 Z-5` with a CLEARANCE of 4  =>  9mm from bed to nozzle.
 *     But: `M851 Z+1` with a CLEARANCE of 2  =>  2mm from bed to nozzle.
 */
#define Z_CLEARANCE_DEPLOY_PROBE   15 // Z Clearance for Deploy/Stow
#define Z_CLEARANCE_BETWEEN_PROBES  5 // Z Clearance between probe points

// For M851 give a range for adjusting the Z probe offset
#define Z_PROBE_OFFSET_RANGE_MIN -20
#define Z_PROBE_OFFSET_RANGE_MAX 20

// Enable the M48 repeatability test to test probe accuracy
//#define Z_MIN_PROBE_REPEATABILITY_TEST

// For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting
(Active High) use 1
// :{ 0:'Low', 1:'High' }
#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 stepper immediately when it's not being used.
// WARNING: When motors turn off there is a chance of losing position
accuracy!
#define DISABLE_X false
#define DISABLE_Y false
#define DISABLE_Z false
// Warn on display about possibly reduced accuracy
//#define DISABLE_REDUCED_ACCURACY_WARNING

// @section extruder

#define DISABLE_E false // For all extruders
#define DISABLE_INACTIVE_EXTRUDER true // Keep only the active extruder
enabled.

// @section machine

// Invert the stepper direction. Change (or reverse the motor connector)
if an axis goes the wrong way.
#define INVERT_X_DIR false // DELTA does not invert
#define INVERT_Y_DIR false
#define INVERT_Z_DIR false

// Enable this option for Toshiba stepper drivers
//#define CONFIG_STEPPERS_TOSHIBA
```

```
// @section extruder

// For direct drive extruder v9 set to true, for geared extruder set to
false.
#define INVERT_E0_DIR false
#define INVERT_E1_DIR false
#define INVERT_E2_DIR false
#define INVERT_E3_DIR false
#define INVERT_E4_DIR false

// @section homing

//#define NO_MOTION_BEFORE_HOMING  // Inhibit movement until all axes
have been homed

//#define Z_HOMING_HEIGHT 4  // (in mm) Minimal z height before homing
(G28) for Z clearance above the bed, clamps, ...
                             // Be sure you have this distance over your
Z_MAX_POS in case.

// Direction of endstops when homing; 1=MAX, -1=MIN
// :[-1,1]
#define X_HOME_DIR 1  // deltas always home to max
#define Y_HOME_DIR 1
#define Z_HOME_DIR 1

// @section machine

// The size of the print bed
#define X_BED_SIZE ((DELTA_PRINTABLE_RADIUS) * 2)
#define Y_BED_SIZE ((DELTA_PRINTABLE_RADIUS) * 2)

// Travel limits (mm) after homing, corresponding to endstop positions.
#define X_MIN_POS -(DELTA_PRINTABLE_RADIUS)
#define Y_MIN_POS -(DELTA_PRINTABLE_RADIUS)
#define Z_MIN_POS 0
#define X_MAX_POS DELTA_PRINTABLE_RADIUS
#define Y_MAX_POS DELTA_PRINTABLE_RADIUS
#define Z_MAX_POS MANUAL_Z_HOME_POS

/**
 * Software Endstops
 *
 * - Prevent moves outside the set machine bounds.
 * - Individual axes can be disabled, if desired.
 * - X and Y only apply to Cartesian robots.
 * - Use 'M211' to set software endstops on/off or report current state
 */

// Min software endstops curtail movement below minimum coordinate bounds
#define MIN_SOFTWARE_ENDSTOPS
#if ENABLED(MIN_SOFTWARE_ENDSTOPS)
  #define MIN_SOFTWARE_ENDSTOP_X
  #define MIN_SOFTWARE_ENDSTOP_Y
  #define MIN_SOFTWARE_ENDSTOP_Z
#endif

// Max software endstops curtail movement above maximum coordinate bounds
#define MAX_SOFTWARE_ENDSTOPS
```

```
#if ENABLED(MAX_SOFTWARE_ENDSTOPS)
  #define MAX_SOFTWARE_ENDSTOP_X
  #define MAX_SOFTWARE_ENDSTOP_Y
  #define MAX_SOFTWARE_ENDSTOP_Z
#endif

/**
 * Filament Runout Sensor
 * A mechanical or opto endstop is used to check for the presence of
filament.
 *
 * RAMPS-based boards use SERVO3_PIN.
 * For other boards you may need to define FIL_RUNOUT_PIN.
 * By default the firmware assumes HIGH = has filament, LOW = ran out
 */
//#define FILAMENT_RUNOUT_SENSOR
#if ENABLED(FILAMENT_RUNOUT_SENSOR)
  #define FIL_RUNOUT_INVERTING false // set to true to invert the logic
of the sensor.
  #define ENDSTOPPULLUP_FIL_RUNOUT // Uncomment to use internal pullup
for filament runout pins if the sensor is defined.
  #define FILAMENT_RUNOUT_SCRIPT "M600"
#endif

//==========================================================================
====
//============================== Bed Leveling
==========================
//==========================================================================
====
// @section calibrate

/**
 * Choose one of the options below to enable G29 Bed Leveling. The
parameters
 * and behavior of G29 will change depending on your selection.
 *
 *  If using a Probe for Z Homing, enable Z_SAFE_HOMING also!
 *
 * - AUTO_BED_LEVELING_3POINT
 *   Probe 3 arbitrary points on the bed (that aren't collinear)
 *   You specify the XY coordinates of all 3 points.
 *   The result is a single tilted plane. Best for a flat bed.
 *
 * - AUTO_BED_LEVELING_LINEAR
 *   Probe several points in a grid.
 *   You specify the rectangle and the density of sample points.
 *   The result is a single tilted plane. Best for a flat bed.
 *
 * - AUTO_BED_LEVELING_BILINEAR
 *   Probe several points in a grid.
 *   You specify the rectangle and the density of sample points.
 *   The result is a mesh, best for large or uneven beds.
 *
 * - AUTO_BED_LEVELING_UBL (Unified Bed Leveling)
 *   A comprehensive bed leveling system combining the features and
benefits
 *   of other systems. UBL also includes integrated Mesh Generation, Mesh
 *   Validation and Mesh Editing systems.
 *
 * - MESH_BED_LEVELING
 *   Probe a grid manually
 *   The result is a mesh, suitable for large or uneven beds. (See
BILINEAR.)
 *   For machines without a probe, Mesh Bed Leveling provides a method to
perform
 *   leveling in steps so you can manually adjust the Z height at each
grid-point.
 *   With an LCD controller the process is guided step-by-step.
 */
//#define AUTO_BED_LEVELING_3POINT
//#define AUTO_BED_LEVELING_LINEAR
//#define AUTO_BED_LEVELING_BILINEAR
//#define AUTO_BED_LEVELING_UBL
//#define MESH_BED_LEVELING

/**
 * Enable detailed logging of G28, G29, M48, etc.
 * Turn on with the command 'M111 S32'.
 * NOTE: Requires a lot of PROGMEM!
 */
//#define DEBUG_LEVELING_FEATURE

#if ENABLED(MESH_BED_LEVELING) || ENABLED(AUTO_BED_LEVELING_BILINEAR) ||
ENABLED(AUTO_BED_LEVELING_UBL)
  // Gradually reduce leveling correction until a set height is reached,
  // at which point movement will be level to the machine's XY plane.
  // The height can be set with M420 Z<height>
  //#define ENABLE_LEVELING_FADE_HEIGHT

  // Set the boundaries for probing (where the probe can reach).
  #define DELTA_PROBEABLE_RADIUS (DELTA_PRINTABLE_RADIUS - 10)


  // For Cartesian machines, instead of dividing moves on mesh
boundaries,
  // split up moves into short segments like a Delta. This follows the
  // contours of the bed more closely than edge-to-edge straight moves.
  #define SEGMENT_LEVELED_MOVES
  #define LEVELED_SEGMENT_LENGTH 5.0 // (mm) Length of all segments
(except the last one)

  /**
   * Enable the G26 Mesh Validation Pattern tool.
   */
  //#define G26_MESH_VALIDATION   // Enable G26 mesh validation
  #if ENABLED(G26_MESH_VALIDATION)
    #define MESH_TEST_NOZZLE_SIZE     0.4   // (mm) Diameter of primary
nozzle.
    #define MESH_TEST_LAYER_HEIGHT    0.2   // (mm) Default layer height
for the G26 Mesh Validation Tool.
    #define MESH_TEST_HOTEND_TEMP   205.0   // (Â°C) Default nozzle
temperature for the G26 Mesh Validation Tool.
    #define MESH_TEST_BED_TEMP       60.0   // (Â°C) Default bed
temperature for the G26 Mesh Validation Tool.
  #endif

#endif
```

```
#if ENABLED(AUTO_BED_LEVELING_LINEAR) ||
ENABLED(AUTO_BED_LEVELING_BILINEAR)

  // Set the number of grid points per dimension.
  // Works best with 5 or more points in each dimension.
  #define GRID_MAX_POINTS_X 9
  #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X

  #define LEFT_PROBE_BED_POSITION -(DELTA_PROBEABLE_RADIUS)
  #define RIGHT_PROBE_BED_POSITION DELTA_PROBEABLE_RADIUS
  #define FRONT_PROBE_BED_POSITION -(DELTA_PROBEABLE_RADIUS)
  #define BACK_PROBE_BED_POSITION DELTA_PROBEABLE_RADIUS

  // The Z probe minimum outer margin (to validate G29 parameters).
  #define MIN_PROBE_EDGE 10

  // Probe along the Y axis, advancing X after each column
  //#define PROBE_Y_FIRST

  #if ENABLED(AUTO_BED_LEVELING_BILINEAR)

    // Beyond the probed grid, continue the implied tilt?
    // Default is to maintain the height of the nearest edge.
    //#define EXTRAPOLATE_BEYOND_GRID

    //
    // Experimental Subdivision of the grid by Catmull-Rom method.
    // Synthesizes intermediate points to produce a more detailed mesh.
    //
    //#define ABL_BILINEAR_SUBDIVISION
    #if ENABLED(ABL_BILINEAR_SUBDIVISION)
      // Number of subdivisions between probe points
      #define BILINEAR_SUBDIVISIONS 3
    #endif

  #endif

#elif ENABLED(AUTO_BED_LEVELING_3POINT)

  // 3 arbitrary points to probe.
  // A simple cross-product is used to estimate the plane of the bed.
  #define ABL_PROBE_PT_1_X 15
  #define ABL_PROBE_PT_1_Y 180
  #define ABL_PROBE_PT_2_X 15
  #define ABL_PROBE_PT_2_Y 20
  #define ABL_PROBE_PT_3_X 170
  #define ABL_PROBE_PT_3_Y 20

#elif ENABLED(AUTO_BED_LEVELING_UBL)


//=========================================================================
====
  //========================= Unified Bed Leveling
============================

//=========================================================================
====
  //#define MESH_EDIT_GFX_OVERLAY   // Display a graphics overlay while
editing the mesh

  #define MESH_INSET 1              // Mesh inset margin on print area
  #define GRID_MAX_POINTS_X 10      // Don't use more than 15 points per
axis, implementation limited.
  #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X

  #define _PX(R,A) (R) * cos(RADIANS(A))
  #define _PY(R,A) (R) * sin(RADIANS(A))
  #define UBL_PROBE_PT_1_X _PX(DELTA_PROBEABLE_RADIUS, 0)   // Probing
points for 3-Point leveling of the mesh
  #define UBL_PROBE_PT_1_Y _PY(DELTA_PROBEABLE_RADIUS, 0)
  #define UBL_PROBE_PT_2_X _PX(DELTA_PROBEABLE_RADIUS, 120)
  #define UBL_PROBE_PT_2_Y _PY(DELTA_PROBEABLE_RADIUS, 120)
  #define UBL_PROBE_PT_3_X _PX(DELTA_PROBEABLE_RADIUS, 240)
  #define UBL_PROBE_PT_3_Y _PY(DELTA_PROBEABLE_RADIUS, 240)

  #define UBL_MESH_EDIT_MOVES_Z     // Sophisticated users prefer no
movement of nozzle
  #define UBL_SAVE_ACTIVE_ON_M500   // Save the currently active mesh in
the current slot on M500

#elif ENABLED(MESH_BED_LEVELING)


//==========================================================================
====
  //=================================== Mesh
==================================

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

  #define MESH_INSET 10          // Mesh inset margin on print area
  #define GRID_MAX_POINTS_X 3    // Don't use more than 7 points per
axis, implementation limited.
  #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X

  //#define MESH_G28_REST_ORIGIN // After homing all axes ('G28' or 'G28
XYZ') rest Z at Z_MIN_POS

#endif // BED_LEVELING

/**
 * Use the LCD controller for bed leveling
 * Requires MESH_BED_LEVELING or PROBE_MANUALLY
 */
//#define LCD_BED_LEVELING

#if ENABLED(LCD_BED_LEVELING)
  #define MBL_Z_STEP 0.025    // Step size while manually probing Z axis.
  #define LCD_PROBE_Z_RANGE 4 // Z Range centered on Z_MIN_POS for LCD Z
adjustment
#endif

// Add a menu item to move between bed corners for manual bed adjustment
//#define LEVEL_BED_CORNERS

/**
```

```
 * Commands to execute at the end of G29 probing.
 * Useful to retract or move the Z probe out of the way.
 */
//#define Z_PROBE_END_SCRIPT "G1 Z10 F12000\nG1 X15 Y330\nG1 Z0.5\nG1
Z10"


// @section homing

// The center of the bed is at (X=0, Y=0)
#define BED_CENTER_AT_0_0

// Manually set the home position. Leave these undefined for automatic
settings.
// For DELTA this is the top-center of the Cartesian print volume.
//#define MANUAL_X_HOME_POS 0
//#define MANUAL_Y_HOME_POS 0
#define MANUAL_Z_HOME_POS DELTA_HEIGHT // Distance between the nozzle to
printbed after homing

// Use "Z Safe Homing" to avoid homing with a Z probe outside the bed
area.
//
// With this feature enabled:
//
// - Allow Z homing only after X and Y homing AND stepper drivers still
enabled.
// - If stepper drivers time out, it will need X and Y homing again
before Z homing.
// - Move the Z probe (or nozzle) to a defined XY point before Z Homing
when homing all axes (G28).
// - Prevent Z homing when the Z probe is outside bed area.
//
//#define Z_SAFE_HOMING

#if ENABLED(Z_SAFE_HOMING)
  #define Z_SAFE_HOMING_X_POINT ((X_BED_SIZE) / 2)    // X point for Z
homing when homing all axes (G28).
  #define Z_SAFE_HOMING_Y_POINT ((Y_BED_SIZE) / 2)    // Y point for Z
homing when homing all axes (G28).
#endif

// Delta only homes to Z
#define HOMING_FEEDRATE_Z  (200*60)

// @section calibrate

/**
 * Bed Skew Compensation
 *
 * This feature corrects for misalignment in the XYZ axes.
 *
 * Take the following steps to get the bed skew in the XY plane:
 *  1. Print a test square (e.g.,
https://www.thingiverse.com/thing:2563185)
 *  2. For XY_DIAG_AC measure the diagonal A to C
 *  3. For XY_DIAG_BD measure the diagonal B to D
 *  4. For XY_SIDE_AD measure the edge A to D
 *
 * Marlin automatically computes skew factors from these measurements.
 * Skew factors may also be computed and set manually:
 *
 *  - Compute AB     : SQRT(2*AC*AC+2*BD*BD-4*AD*AD)/2
 *  - XY_SKEW_FACTOR : TAN(PI/2-ACOS((AC*AC-AB*AB-AD*AD)/(2*AB*AD)))
 *
 * If desired, follow the same procedure for XZ and YZ.
 * Use these diagrams for reference:
 *
 *    Y                     Z                     Z
 *    ^     B-------C       ^     B-------C       ^     B-------C
 *    |    /       /        |    /       /        |    /       /
 *    |   /       /         |   /       /         |   /       /
 *    |  A-------D          |  A-------D          |  A-------D
 *    +-------------->X     +-------------->X     +-------------->Y
 *     XY_SKEW_FACTOR        XZ_SKEW_FACTOR        YZ_SKEW_FACTOR
 */
//#define SKEW_CORRECTION

#if ENABLED(SKEW_CORRECTION)
  // Input all length measurements here:
  #define XY_DIAG_AC 282.8427124746
  #define XY_DIAG_BD 282.8427124746
  #define XY_SIDE_AD 200

  // Or, set the default skew factors directly here
  // to override the above measurements:
  #define XY_SKEW_FACTOR 0.0

  //#define SKEW_CORRECTION_FOR_Z
  #if ENABLED(SKEW_CORRECTION_FOR_Z)
    #define XZ_DIAG_AC 282.8427124746
    #define XZ_DIAG_BD 282.8427124746
    #define YZ_DIAG_AC 282.8427124746
    #define YZ_DIAG_BD 282.8427124746
    #define YZ_SIDE_AD 200
    #define XZ_SKEW_FACTOR 0.0
    #define YZ_SKEW_FACTOR 0.0
  #endif

  // Enable this option for M852 to set skew at runtime
  //#define SKEW_CORRECTION_GCODE
#endif

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

// @section extras

//
// 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 EEPROM_SETTINGS // Enable for M500 and M501 commands
//#define DISABLE_M503    // Saves ~2700 bytes of PROGMEM. Disable for
release!
#define EEPROM_CHITCHAT   // Give feedback on EEPROM commands. Disable to
save PROGMEM.

//
// Host Keepalive
//
// When enabled Marlin will send a busy status message to the host
// every couple of seconds when it can't accept commands.
//
#define HOST_KEEPALIVE_FEATURE        // Disable this if your host
doesn't like keepalive messages
#define DEFAULT_KEEPALIVE_INTERVAL 2  // Number of seconds between "busy"
messages. Set with M113.
#define BUSY_WHILE_HEATING            // Some hosts require "busy"
messages even during heating

//
// M100 Free Memory Watcher
//
//#define M100_FREE_MEMORY_WATCHER    // Add M100 (Free Memory Watcher)
to debug memory usage

//
// G20/G21 Inch mode support
//
//#define INCH_MODE_SUPPORT

//
// M149 Set temperature units support
//
//#define TEMPERATURE_UNITS_SUPPORT

// @section temperature

// Preheat Constants
#define PREHEAT_1_TEMP_HOTEND 180
#define PREHEAT_1_TEMP_BED     70
#define PREHEAT_1_FAN_SPEED   255 // Value from 0 to 255

#define PREHEAT_2_TEMP_HOTEND 240
#define PREHEAT_2_TEMP_BED    100
#define PREHEAT_2_FAN_SPEED   255 // Value from 0 to 255

/**
 * Nozzle Park
 *
 * Park the nozzle at the given XYZ position on idle or G27.
 *
 * The "P" parameter controls the action applied to the Z axis:
 *
 *    P0  (Default) If Z is below park Z raise the nozzle.
 *    P1  Raise the nozzle always to Z-park height.
 *    P2  Raise the nozzle by Z-park amount, limited to Z_MAX_POS.
 */
```

```
//#define NOZZLE_PARK_FEATURE

#if ENABLED(NOZZLE_PARK_FEATURE)
  // Specify a park position as { X, Y, Z }
  #define NOZZLE_PARK_POINT { (X_MIN_POS + 10), (Y_MAX_POS - 10), 20 }
  #define NOZZLE_PARK_XY_FEEDRATE 100   // X and Y axes feedrate in mm/s
(also used for delta printers Z axis)
  #define NOZZLE_PARK_Z_FEEDRATE 5      // Z axis feedrate in mm/s (not
used for delta printers)
#endif

/**
 * Clean Nozzle Feature -- EXPERIMENTAL
 *
 * Adds the G12 command to perform a nozzle cleaning process.
 *
 * Parameters:
 *   P  Pattern
 *   S  Strokes / Repetitions
 *   T  Triangles (P1 only)
 *
 * Patterns:
 *   P0  Straight line (default). This process requires a sponge type
material
 *       at a fixed bed location. "S" specifies strokes (i.e. back-forth
motions)
 *       between the start / end points.
 *
 *   P1  Zig-zag pattern between (X0, Y0) and (X1, Y1), "T" specifies the
 *       number of zig-zag triangles to do. "S" defines the number of
strokes.
 *       Zig-zags are done in whichever is the narrower dimension.
 *       For example, "G12 P1 S1 T3" will execute:
 *
 *          --
 *         |  (X0, Y1) |     /\        /\        /\     | (X1, Y1)
 *         |           |    /  \      /  \      /  \    |
 *       A |           |   /    \    /    \    /    \   |
 *         |           |  /      \  /      \  /      \  |
 *         |  (X0, Y0) | /        \/        \/        \ | (X1, Y0)
 *          --         +--------------------------------+
 *                       |________|_________|_________|
 *                           T1        T2        T3
 *
 *   P2  Circular pattern with middle at NOZZLE_CLEAN_CIRCLE_MIDDLE.
 *       "R" specifies the radius. "S" specifies the stroke count.
 *       Before starting, the nozzle moves to NOZZLE_CLEAN_START_POINT.
 *
 *   Caveats: The ending Z should be the same as starting Z.
 * Attention: EXPERIMENTAL. G-code arguments may change.
 *
 */
//#define NOZZLE_CLEAN_FEATURE

#if ENABLED(NOZZLE_CLEAN_FEATURE)
  // Default number of pattern repetitions
  #define NOZZLE_CLEAN_STROKES  12

  // Default number of triangles
  #define NOZZLE_CLEAN_TRIANGLES  3
  // Specify positions as { X, Y, Z }
  #define NOZZLE_CLEAN_START_POINT { 30, 30, (Z_MIN_POS + 1)}
  #define NOZZLE_CLEAN_END_POINT   {100, 60, (Z_MIN_POS + 1)}

  // Circular pattern radius
  #define NOZZLE_CLEAN_CIRCLE_RADIUS 6.5
  // Circular pattern circle fragments number
  #define NOZZLE_CLEAN_CIRCLE_FN 10
  // Middle point of circle
  #define NOZZLE_CLEAN_CIRCLE_MIDDLE NOZZLE_CLEAN_START_POINT

  // Moves the nozzle to the initial position
  #define NOZZLE_CLEAN_GOBACK
#endif

/**
 * Print Job Timer
 *
 * Automatically start and stop the print job timer on M104/M109/M190.
 *
 *   M104 (hotend, no wait) - high temp = none,        low temp = stop
timer
 *   M109 (hotend, wait)    - high temp = start timer, low temp = stop
timer
 *   M190 (bed, wait)       - high temp = start timer, low temp = none
 *
 * The timer can also be controlled with the following commands:
 *
 *   M75 - Start the print job timer
 *   M76 - Pause the print job timer
 *   M77 - Stop the print job timer
 */
#define PRINTJOB_TIMER_AUTOSTART

/**
 * Print Counter
 *
 * Track statistical data such as:
 *
 *  - Total print jobs
 *  - Total successful print jobs
 *  - Total failed print jobs
 *  - Total time printing
 *
 * View the current statistics with M78.
 */
//#define PRINTCOUNTER

//==========================================================================
======
//============================= LCD and SD support
==========================
//==========================================================================
======

// @section lcd

/**
 * LCD LANGUAGE
 *
 * Select the language to display on the LCD. These languages are
available:
 *
 *    en, an, bg, ca, cn, cz, cz_utf8, de, el, el-gr, es, eu, fi, fr,
fr_utf8, gl,
 *    hr, it, kana, kana_utf8, nl, pl, pt, pt_utf8, pt-br, pt-br_utf8,
ru, sk_utf8,
 *    tr, uk, zh_CN, zh_TW, test
 *
 * :{ 'en':'English', 'an':'Aragonese', 'bg':'Bulgarian', 'ca':'Catalan',
'cn':'Chinese', 'cz':'Czech', 'cz_utf8':'Czech (UTF8)', 'de':'German',
'el':'Greek', 'el-gr':'Greek (Greece)', 'es':'Spanish', 'eu':'Basque-
Euskera', 'fi':'Finnish', 'fr':'French', 'fr_utf8':'French (UTF8)',
'gl':'Galician', 'hr':'Croatian', 'it':'Italian', 'kana':'Japanese',
'kana_utf8':'Japanese (UTF8)', 'nl':'Dutch', 'pl':'Polish',
'pt':'Portuguese', 'pt-br':'Portuguese (Brazilian)', 'pt-
br_utf8':'Portuguese (Brazilian UTF8)', 'pt_utf8':'Portuguese (UTF8)',
'ru':'Russian', 'sk_utf8':'Slovak (UTF8)', 'tr':'Turkish',
'uk':'Ukrainian', 'zh_CN':'Chinese (Simplified)', 'zh_TW':'Chinese
(Taiwan)', test':'TEST' }
 */
#define LCD_LANGUAGE en

/**
 * LCD Character Set
 *
 * Note: This option is NOT applicable to Graphical Displays.
 *
 * All character-based LCDs provide ASCII plus one of these
 * language extensions:
 *
 *  - JAPANESE ... the most common
 *  - WESTERN  ... with more accented characters
 *  - CYRILLIC ... for the Russian language
 *
 * To determine the language extension installed on your controller:
 *
 *  - Compile and upload with LCD_LANGUAGE set to 'test'
 *  - Click the controller to view the LCD menu
 *  - The LCD will display Japanese, Western, or Cyrillic text
 *
 * See http://marlinfw.org/docs/development/lcd_language.html
 *
 * :['JAPANESE', 'WESTERN', 'CYRILLIC']
 */
#define DISPLAY_CHARSET_HD44780 JAPANESE

/**
 * LCD TYPE
 *
 * Enable ULTRA_LCD for a 16x2, 16x4, 20x2, or 20x4 character-based LCD.
 * Enable DOGLCD for a 128x64 (ST7565R) Full Graphical Display.
 * (These options will be enabled automatically for most displays.)
 *
 * IMPORTANT: The U8glib library is required for Full Graphic Display!
 *            https://github.com/olikraus/U8glib_Arduino
 */
//#define ULTRA_LCD   // Character based
//#define DOGLCD      // Full graphics display
```

```
/**
 * SD CARD
 *
 * SD Card support is disabled by default. If your controller has an SD
slot,
 * you must uncomment the following option or it won't work.
 *
 */
#define SDSUPPORT

/**
 * SD CARD: SPI SPEED
 *
 * Enable one of the following items for a slower SPI transfer speed.
 * This may be required to resolve "volume init" errors.
 */
//#define SPI_SPEED SPI_HALF_SPEED
//#define SPI_SPEED SPI_QUARTER_SPEED
//#define SPI_SPEED SPI_EIGHTH_SPEED

/**
 * SD CARD: ENABLE CRC
 *
 * Use CRC checks and retries on the SD communication.
 */
//#define SD_CHECK_AND_RETRY

//
// ENCODER SETTINGS
//
// This option overrides the default number of encoder pulses needed to
// produce one step. Should be increased for high-resolution encoders.
//
//#define ENCODER_PULSES_PER_STEP 1

//
// Use this option to override the number of step signals required to
// move between next/prev menu items.
//
//#define ENCODER_STEPS_PER_MENU_ITEM 5

/**
 * Encoder Direction Options
 *
 * Test your encoder's behavior first with both options disabled.
 *
 *  Reversed Value Edit and Menu Nav? Enable REVERSE_ENCODER_DIRECTION.
 *  Reversed Menu Navigation only?    Enable REVERSE_MENU_DIRECTION.
 *  Reversed Value Editing only?      Enable BOTH options.
 */

//
// This option reverses the encoder direction everywhere.
//
//  Set this option if CLOCKWISE causes values to DECREASE
//
//#define REVERSE_ENCODER_DIRECTION

//
```

```
// This option reverses the encoder direction for navigating LCD menus.
//
//  If CLOCKWISE normally moves DOWN this makes it go UP.
//  If CLOCKWISE normally moves UP this makes it go DOWN.
//
//#define REVERSE_MENU_DIRECTION

//
// Individual Axis Homing
//
// Add individual axis homing items (Home X, Home Y, and Home Z) to the
LCD menu.
//
//#define INDIVIDUAL_AXIS_HOMING_MENU

//
// SPEAKER/BUZZER
//
// If you have a speaker that can produce tones, enable it here.
// By default Marlin assumes you have a buzzer with a fixed frequency.
//
//#define SPEAKER

//
// The duration and frequency for the UI feedback sound.
// Set these to 0 to disable audio feedback in the LCD menus.
//
// Note: Test audio output with the G-Code:
//  M300 S<frequency Hz> P<duration ms>
//
//#define LCD_FEEDBACK_FREQUENCY_DURATION_MS 2
//#define LCD_FEEDBACK_FREQUENCY_HZ 5000

//
// CONTROLLER TYPE: Standard
//
// Marlin supports a wide variety of controllers.
// Enable one of the following options to specify your controller.
//

//
// ULTIMAKER Controller.
//
//#define ULTIMAKERCONTROLLER

//
// ULTIPANEL as seen on Thingiverse.
//
//#define ULTIPANEL

//
// PanelOne from T3P3 (via RAMPS 1.4 AUX2/AUX3)
// http://reprap.org/wiki/PanelOne
//
//#define PANEL_ONE

//
// MaKr3d Makr-Panel with graphic controller and SD support.
// http://reprap.org/wiki/MaKr3d_MaKrPanel
//
```

```
//#define MAKRPANEL

//
// ReprapWorld Graphical LCD
// https://reprapworld.com/?products_details&products_id/1218
//
//#define REPRAPWORLD_GRAPHICAL_LCD

//
// Activate one of these if you have a Panucatt Devices
// Viki 2.0 or mini Viki with Graphic LCD
// http://panucatt.com
//
//#define VIKI2
//#define miniVIKI

//
// Adafruit ST7565 Full Graphic Controller.
// https://github.com/eboston/Adafruit-ST7565-Full-Graphic-Controller/
//
//#define ELB_FULL_GRAPHIC_CONTROLLER

//
// RepRapDiscount Smart Controller.
// http://reprap.org/wiki/RepRapDiscount_Smart_Controller
//
// Note: Usually sold with a white PCB.
//
//#define REPRAP_DISCOUNT_SMART_CONTROLLER

//
// GADGETS3D G3D LCD/SD Controller
// http://reprap.org/wiki/RAMPS_1.3/1.4_GADGETS3D_Shield_with_Panel
//
// Note: Usually sold with a blue PCB.
//
//#define G3D_PANEL

//
// RepRapDiscount FULL GRAPHIC Smart Controller
// http://reprap.org/wiki/RepRapDiscount_Full_Graphic_Smart_Controller
//
#define REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER

//
// MakerLab Mini Panel with graphic
// controller and SD support - http://reprap.org/wiki/Mini_panel
//
//#define MINIPANEL

//
// RepRapWorld REPRAPWORLD_KEYPAD v1.1
//
http://reprapworld.com/?products_details&products_id=202&cPath=1591_1626
//
// REPRAPWORLD_KEYPAD_MOVE_STEP sets how much should the robot move when
a key
// is pressed, a value of 10.0 means 10mm per click.
//
//#define REPRAPWORLD_KEYPAD
```

```
//#define REPRAPWORLD_KEYPAD_MOVE_STEP 1.0

//
// RigidBot Panel V1.0
// http://www.inventapart.com/
//
//#define RIGIDBOT_PANEL

//
// BQ LCD Smart Controller shipped by
// default with the BQ Hephestos 2 and Witbox 2.
//
//#define BQ_LCD_SMART_CONTROLLER

//
// Cartesio UI
// http://mauk.cc/webshop/cartesio-shop/electronics/user-interface
//
//#define CARTESIO_UI

//
// ANET and Tronxy Controller supported displays.
//
//#define ZONESTAR_LCD            // Requires ADC_KEYPAD_PIN to be
assigned to an analog pin.
                                  // This LCD is known to be susceptible
to electrical interference
                                  // which scrambles the display.
Pressing any button clears it up.
                                  // This is a LCD2004 display with 5
analog buttons.

//#define ANET_FULL_GRAPHICS_LCD  // Anet 128x64 full graphics lcd with
rotary encoder as used on Anet A6
                                  // A clone of the RepRapDiscount full
graphics display but with
                                  // different pins/wiring (see
pins_ANET_10.h).

//
// LCD for Melzi Card with Graphical LCD
//
//#define LCD_FOR_MELZI

//
// CONTROLLER TYPE: I2C
//
// Note: These controllers require the installation of Arduino's
LiquidCrystal_I2C
// library. For more info:
https://github.com/kiyoshigawa/LiquidCrystal_I2C
//

//
// Elefu RA Board Control Panel
// http://www.elefu.com/index.php?route=product/product&product_id=53
//
//#define RA_CONTROL_PANEL

//
```

```
// Sainsmart YW Robot (LCM1602) LCD Display
//
// Note: This controller requires F.Malpartida's LiquidCrystal_I2C
library
// https://bitbucket.org/fmalpartida/new-liquidcrystal/wiki/Home
//
//#define LCD_I2C_SAINSMART_YWROBOT

//
// Generic LCM1602 LCD adapter
//
//#define LCM1602

//
// PANELOLU2 LCD with status LEDs,
// separate encoder and click inputs.
//
// Note: This controller requires Arduino's LiquidTWI2 library v1.2.3 or
later.
// For more info: https://github.com/lincomatic/LiquidTWI2
//
// 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_PANELOLU2

//
// Panucatt VIKI LCD with status LEDs,
// integrated click & L/R/U/D buttons, separate encoder inputs.
//
//#define LCD_I2C_VIKI

//
// SSD1306 OLED full graphics generic display
//
//#define U8GLIB_SSD1306

//
// SAV OLEd LCD module support using either SSD1306 or SH1106 based LCD
modules
//
//#define SAV_3DGLCD
#if ENABLED(SAV_3DGLCD)
  //#define U8GLIB_SSD1306
  #define U8GLIB_SH1106
#endif

//
// CONTROLLER TYPE: Shift register panels
//
// 2 wire Non-latching LCD SR from https://goo.gl/aJJ4sH
// LCD configuration: http://reprap.org/wiki/SAV_3D_LCD
//
//#define SAV_3DLCD

//
// TinyBoy2 128x64 OLED / Encoder Panel
//
```

```
//#define OLED_PANEL_TINYBOY2

//
// Makeboard 3D Printer Parts 3D Printer Mini Display 1602 Mini
Controller
// https://www.aliexpress.com/item/Micromake-Makeboard-3D-Printer-Parts-
3D-Printer-Mini-Display-1602-Mini-Controller-Compatible-with-Ramps-
1/32765887917.html
//
//#define MAKEBOARD_MINI_2_LINE_DISPLAY_1602

//
// MKS MINI12864 with graphic controller and SD support
// http://reprap.org/wiki/MKS_MINI_12864
//
//#define MKS_MINI_12864

//
// Factory display for Creality CR-10
// https://www.aliexpress.com/item/Universal-LCD-12864-3D-Printer-
Display-Screen-With-Encoder-For-CR-10-CR-7-Model/32833148327.html
//
// This is RAMPS-compatible using a single 10-pin connector.
// (For CR-10 owners who want to replace the Melzi Creality board but
retain the display)
//
//#define CR10_STOCKDISPLAY

//
// MKS OLED 1.3" 128 Ã— 64 FULL GRAPHICS CONTROLLER
// http://reprap.org/wiki/MKS_12864OLED
//
// Tiny, but very sharp OLED display
//
//#define MKS_12864OLED          // Uses the SH1106 controller (default)
//#define MKS_12864OLED_SSD1306  // Uses the SSD1306 controller

// Silvergate GLCD controller
// http://github.com/android444/Silvergate
//
//#define SILVER_GATE_GLCD_CONTROLLER

//=========================================================================
======
//============================== Extra Features
==============================
//=========================================================================
======

// @section extras

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

// Use software PWM to drive the fan, as for the heaters. This uses a
very low frequency
// which is not as 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

// If SOFT_PWM_SCALE is set to a value higher than 0, dithering can
// be used to mitigate the associated resolution loss. If enabled,
// some of the PWM cycles are stretched so on average the desired
// duty cycle is attained.
//#define SOFT_PWM_DITHER

// Temperature status LEDs that display the hotend and bed temperature.
// If all hotends, bed temperature, and target temperature are under 54C
// then the BLUE led is on. Otherwise the RED led is on. (1C hysteresis)
//#define TEMP_STAT_LEDS

// 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

// SkeinForge sends the wrong arc g-codes when using Arc Point as fillet
procedure
//#define SF_ARC_FIX

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

// Support for BlinkM/CyzRgb
//#define BLINKM

// Support for PCA9632 PWM LED driver
//#define PCA9632

/**
 * RGB LED / LED Strip Control
 *
 * Enable support for an RGB LED connected to 5V digital pins, or
 * an RGB Strip connected to MOSFETs controlled by digital pins.
 *
 * Adds the M150 command to set the LED (or LED strip) color.
 * If pins are PWM capable (e.g., 4, 5, 6, 11) then a range of
 * luminance values can be set from 0 to 255.
 * For Neopixel LED an overall brightness parameter is also available.
 *
 * *** CAUTION ***
 *  LED Strips require a MOFSET Chip between PWM lines and LEDs,
 *  as the Arduino cannot handle the current the LEDs will require.
 *  Failure to follow this precaution can destroy your Arduino!
 *  NOTE: A separate 5V power supply is required! The Neopixel LED needs
 *  more current than the Arduino 5V linear regulator can produce.
 * *** CAUTION ***
 *
 * LED Type. Enable only one of the following two options.
 *
 */
//#define RGB_LED
//#define RGBW_LED
```

```
#if ENABLED(RGB_LED) || ENABLED(RGBW_LED)
  #define RGB_LED_R_PIN 34
  #define RGB_LED_G_PIN 43
  #define RGB_LED_B_PIN 35
  #define RGB_LED_W_PIN -1
#endif

// Support for Adafruit Neopixel LED driver
//#define NEOPIXEL_LED
#if ENABLED(NEOPIXEL_LED)
  #define NEOPIXEL_TYPE   NEO_GRBW // NEO_GRBW / NEO_GRB - four/three
channel driver type (defined in Adafruit_NeoPixel.h)
  #define NEOPIXEL_PIN    4        // LED driving pin on motherboard 4 =>
D4 (EXP2-5 on Printrboard) / 30 => PC7 (EXP3-13 on Rumba)
  #define NEOPIXEL_PIXELS 30       // Number of LEDs in the strip
  #define NEOPIXEL_IS_SEQUENTIAL   // Sequential display for temperature
change - LED by LED. Disable to change all LEDs at once.
  #define NEOPIXEL_BRIGHTNESS 127  // Initial brightness (0-255)
  //#define NEOPIXEL_STARTUP_TEST  // Cycle through colors at startup
#endif

/**
 * Printer Event LEDs
 *
 * During printing, the LEDs will reflect the printer status:
 *
 *  - Gradually change from blue to violet as the heated bed gets to
target temp
 *  - Gradually change from violet to red as the hotend gets to
temperature
 *  - Change to white to illuminate work surface
 *  - Change to green once print has finished
 *  - Turn off after the print has finished and the user has pushed a
button
 */
#if ENABLED(BLINKM) || ENABLED(RGB_LED) || ENABLED(RGBW_LED) ||
ENABLED(PCA9632) || ENABLED(NEOPIXEL_LED)
  #define PRINTER_EVENT_LEDS
#endif

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

/**
 * Number of servos
 *
 * For some servo-related options NUM_SERVOS will be set automatically.
 * Set this manually if there are extra servos needing manual control.
 * Leave undefined or set to 0 to entirely disable the servo subsystem.
 */
//#define NUM_SERVOS 3 // Servo index starts with 0 for M280 command

// Delay (in milliseconds) before the next move will start, to give the
servo time to reach its target angle.
// 300ms is a good value but you can try less delay.
// If the servo can't reach the requested position, increase it.
#define SERVO_DELAY { 300 }
```

```
// Servo deactivation
//
// With this option servos are powered only during movement, then turned
off to prevent jitter.
//#define DEACTIVATE_SERVOS_AFTER_MOVE

#endif // CONFIGURATION_H
```