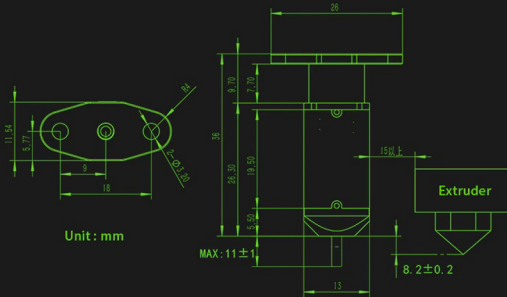


1

3D Touch parameters

Voltage	5V
Current	15mA
Max current(peak)	300mA
Shell material	PC
Shell color	White translucent
Cable length	short:200mm Long: 1000mm
Wire	3pin: Green:-GND Red:+5V Orange: Signal cable 2pin: Black:-GND White: Zmin

Size



2

Action instruction table

3D Touch Action Instruction Table			
Action	G-code		
	Marlin Firmware	Repetier Firmware	Smoothieware Firmware
Push-pin down	M280 P0 S10	M340 P0 S700	M280 S3.0
Push-pin up	M280 P0 S90	M340 P0 S1500	M280 S7.0
Self-test	M280 P0 S120	M340 P0 S1800	M280 S8.4
Alarm release(push-pin up)	M280 P0 S160	M340 P0 S2200	M280 S10.6

3

Marlin settings

Marlin settings (take Marlin V1.1.X as example) 3pin cable connects to D11, 2pin cable connects to Zmin.

(1) Set leveling pin, which can't use the same pin as endstop.

Generally, on Delta machines, we use Zmin pin for BLtouch, Zmax pin for endstop. On I3 machines, we use Zmin pin as BLtouch, Zmax pin for endstop or don't use endstop.

```

// #define DELTA_DIAGONAL_ROD_TRIM_TOWER {0, 0, 0}

#endif

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

// @section homing

// Specify here all the endstop connectors that are connected to any endstop
// Almost all printers will be using one per axis. Probes will use one on
// extra connectors. Leave undefined any used for non-endstop and non-probe

// #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
  
```

4

(2) Set leveling type

```

// #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 false // set to true to invert the logic of the endstop.
#define Y_MIN_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
#define Z_MIN_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
#define X_MAX_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
#define Y_MAX_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
#define Z_MAX_ENDSTOP_INVERTING false // set to true to invert the logic of the endstop.
#define Z_MIN_PROBE_ENDSTOP_INVERTING false // 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
  
```

5

(3) Enable Z_MIN_PROBE_USES_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
// #define Z_MIN_PROBE_PIN Z_MAX_PIN
  
```

6

(3) Enable Z_MIN_PROBE_USES_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
// #define Z_MIN_PROBE_PIN Z_MAX_PIN
  
```

7

(4) Enable BLTouch

```

* 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 100 // (ms) Enable and increase if needed
#endif

/**
 * Enable if probing seems unreliable. Heaters and/or fans - consistent with the
 * options selected below - will be disabled during probing so as to minimize

```

8

(5) Set the offset value of the leveling switch probe from the nozzle in the XYZ direction

```

* T | | H
* | (-) | T
* | |
* O-- FRONT --+
* (0,0)

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

#define X_PROBE_OFFSET_FROM_EXTRUDER 0 // X offset: -left +right [of the nozzle]
#define Y_PROBE_OFFSET_FROM_EXTRUDER -20 // Y offset: -front +behind [the nozzle]
#define Z_PROBE_OFFSET_FROM_EXTRUDER -1.8 // Z offset: -below +above [the nozzle]

```

9

(6) Set leveling mode

```

* 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_UDEL
// #define MESH_BED_LEVELING

```

10

(7) Set leveling points number

```

#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 5
  #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X

```

11

(8) Set homing center

```

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

//
#define Z_SAFE_HOMING

```

12

(9) Store the leveling settings

1) Remove "//" before the "#define EEPROM_SETTINGS", start M500 to store the data.

```

//
// 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)
// M502 - reverts to the default "factory settings". You still need to store them in EEPROM!
//define this to enable EEPROM support
#define EEPROM_SETTINGS

#if ENABLED(EEPROM_SETTINGS)
  // To disable EEPROM Serial responses and decrease program space by ~1700 bytes: comment
  // #define EEPROM_CHITCHAT // Please keep turned on if you can.
#endif

```

13

1) Open Marlin_main.cpp, find "case 28: //G28: Home all axes, one at a time gcode_G28(false);", add this sentence: "set_bed_leveling_enabled(true);". It's important to add this sentence, otherwise the leveling data can't be stored.

```

// #if ENABLED(NOZZLE_PARK_FEATURE)
//   case 27: // G27: Nozzle Park
//     gcode_G27();
//     break;
// #endif // NOZZLE_PARK_FEATURE

// case 28: // G28: Home all axes, one at a time
//   gcode_G28(false);
//   set_bed_leveling_enabled(true);
//   break;

```

14

(10) Remove "//" before the "#define NUM_SERVOS 3"

```

// Number of servos
//
// If you select a configuration below, this will receive a default value and
// set it manually if you have more servos than extruders and wish to manually
// leaving it undefined or defining as 0 will disable the servo subsystem
// If unsure, leave commented / disabled
//
#define NUM_SERVOS 3 // Servo index starts with 0 for M260 command

```