Установка Cartographer3D на QIDI Plus 4 (прошивка 1.6.0)

ДИСКЛЕЙМЕР! Данная модификация не является рекомендацией или призывом к действию, а инструкция только для тех, кому очень хочется использовать датчик карты стола типа Cartographer3D.

Все изменения вы делаете на свой страх и риск и под личную ответственность!!!

Прежде всего, вам понадобится зонд Cartographer3D. Версия USB. Если умеете паять, то можно взять не спаянную версию и собрать любую удобную конфигурацию. Если не умеете паять, то стоит взять версию 90 градусов.

image.png

Я использовал модель крепления:

https://www.printables.com/model/1170120-beacon3d-mount-for-qidi-plus4 **С ней заводская крышка не подходит**

Есть и другие модели, выбирать вам.

Главное в модели крепления вам следует знать смещение центра катушки по осям X и Y относительно центра сопла. Это понадобится позже.

Необходимо убедиться, что катушка находится на расстоянии 2,6–3,0 мм от кончика сопла.

Вам также необходимо протянуть USB-кабель от материнской платы к голове, для этого придется снять заднюю крышку принтера и проложить кабель соответствующим образом.

Комплектный кабель Cartographer3D замечательно помещается в заводской кабель-канал рядом со штатным проводом головы.

image.png

В идеале вы должны использовать один из портов USB 2.0, который является либо средним, либо нижним портом на материнской плате. Верхний порт, в который вставлен Wifi-адаптер, является USB 3.0 (порт синего цвета) и у некоторых пользователей происходит отвал Carto в этом порту.

1. Обновление Python

Cartographer нужен плагин для работы Klipper, этот плагин поддерживается только на Python 3.9 или выше. Qidi Plus 4 поставляется с Python 3.7 по умолчанию, поэтому нам нужно обновится до Python 3.12 и перезагрузить виртуальную среду Klipper.

Далее все действия выполняем через SSH. Что бы зайти в командной строке вводим:

ssh mks@«IP вашего принтера» без скобок

пароль **makerbase** (при вводе пароля буквы не отобразятся, это нормально)

Сначала загрузите и распакуйте предварительно скомпилированный Python 3.12.3 введя:

> sudo service klipper stop
>
> cd ~
>
> wget https://github.com/stew675/ShakeTune_For_Qidi/releases/download/v1.0.0/python-3-12-3.tgz
>
> tar xvzf python-3-12-3.tgz
>
> rm python-3-12-3.tgz

Теперь Python 3.12.3 находится в домашнем каталоге.

2. Мы удалим виртуальную среду Klipper и создадим ее заново.

Не волнуйтесь, это не затронет никакие ваши пользовательские данные, но **после этого вам нужно будет переустановить все плагины klipper.**

```
sudo rm -rf klippy-env

~/python-3.12.3/bin/python3.12 -m venv klippy-env

cd ~/klippy-env

sed -i 's/greenlet==2.0.2/greenlet==3.0.3/' ../klipper/scripts/klippy-requirements.txt # Need to upgrade this package for 3.12.

bin/pip install -r ../klipper/scripts/klippy-requirements.txt
```

3. Установка Cartographer для Klipper

Теперь Klipper установлен снова с использует Python 3.12.

Теперь мы можем установить плагин Cartographer.

```
cd ~

git clone https://github.com/Cartographer3D/cartographer-klipper.git

./cartographer-klipper/install.sh
```

Плагин установлен. И мы можем снова запустить Klipper.

```
sudo service klipper start
```

4. Патч Клиппера

Версия Klipper от Qidi представляет собой модифицированную версию **probe.py** и не будет работать с Cartographer, нам нужно установить патч, чтобы он заработал.

Останавливаем клиппер:

```
```

> sudo service klipper stop

Запускаем команду, которая загрузит исправленную копию **probe.py.**

> wget -O /home/mks/klipper/klippy/extras/probe.py https://raw.githubusercontent.com/qidi-community/Plus4-Wiki/refs/heads/main/content/bed-scanning-probes/Beacon3D/RevH-Normal/probe.py

После этого запустите klipper:

> sudo service klipper start

или выключите и включите принтер.

Сейчас самое время переустановить все плагины Klipper, которые вы использовали. Если вы хотите снова установить **Shake&Tune**, используйте **основную версию**, так как Python теперь имеет более новую версию.

5. Конфигурации Klipper

При перезапуске Klipper, вероятно, произошел сбой с ошибками о том, что что-то не найдено.

image.png

Не волнуйтесь, нам просто нужно обновить файлы конфигурации.

5.1 Изменения в printer.cfg

Нам необходимо внести следующие изменения в **`[Z_stepper]`:**

> `[stepper_z]`

```
step_pin:U_1:PB1

dir_pin:U_1:PB6

enable_pin:!U_1:PB0

microsteps: 16

rotation_distance: 4

full_steps_per_rotation: 200

endstop_pin:probe:z_virtual_endstop # U_1:PC3 for Z-max

endstop_pin_reverse:tmc2209_stepper_z:virtual_endstop

#position_endstop:1

position_endstop_reverse:285

position_max:285

position_min: -4

homing_speed: 10

homing_speed_reverse: 10

second_homing_speed: 5

homing_retract_dist: 0

homing_positive_dir:false

homing_positive_dir_reverse:true

#step_pulse_duration:0.0000001
```

Закомментируйте (или удалите) все строки в следующих разделах, они будут добавлены позже:

> #[z_tilt]
>
> #z_positions:
>
> # -17.5,138.5

```
#    335.7,138.5
#
#points:
#    0,138.5
#
#speed: 150
#horizontal_move_z: 5
#retries: 2
#retry_tolerance: 0.05


#[smart_effector]
#pin:U_1:PC1
#recovery_time:0
#x_offset: 25
#y_offset: 1.3
#z_offset: 0.000001
#speed:5
#lift_speed:5
#probe_accel:50
#samples: 2
#samples_result: submaxmin
#sample_retract_dist: 5
#samples_tolerance: 0.05
#samples_tolerance_retries:5
```

```
#[qdprobe]
#pin:!PA10
#z_offset:0.000001

#[bed_mesh]
#speed:150
#horizontal_move_z:5
#mesh_min:25,10
#mesh_max:295,295
#probe_count:9,9
#algorithm:bicubic
#bicubic_tension:0.4
#mesh_pps: 2,2
```

5.2 Изменения в gcode_macro.cfg

Нам необходимо изменить действия принтера в начале печати, чтобы вместо стандартных датчиков Qidi использовать датчики Carto.

Найдите и замените `[gcode_macro PRINT_START]` на это:

```
[gcode_macro PRINT_START]
gcode:
    AUTOTUNE_SHAPERS
```

```
{% set bedtemp = params.BED|int %}

{% set hotendtemp = params.HOTEND|int %}

{% set chambertemp = params.CHAMBER|default(0)|int %}

SET_GCODE_OFFSET Z=0 # Zero out z-offset

M104 S0

M106 P2 S0

M106 P3 S0

M106 S255

G28

M141 S{chambertemp}

M140 S{bedtemp}

M106 S0

CLEAR_NOZZLE HOTEND={hotendtemp}

M191 S{chambertemp}

M190 S{bedtemp}

M104 S140

G29

G0 Z50 F600

G0 X5 Y5  F6000

{% if chambertemp == 0 %}

   M106 P3 S255

{% endif %}

M109 S{hotendtemp}

M141 S{chambertemp}
```

```
    M204 S10000

    SET_PRINT_STATS_INFO CURRENT_LAYER=1

    ENABLE_ALL_SENSOR

    save_last_file
```

Найдите и замените `[homing_override]` на это:

```
[homing_override]
  axes:xyz
  gcode:
    {% set HOME_CUR = 1 %}
    {% set driver_config = printer.configfile.settings['tmc2240 stepper_x'] %}
    {% set RUN_CUR = driver_config.run_current %}
    {% set HOLD_CUR = driver_config.hold_current %}
    m204 S10000
    M220 S100
    {% if params.X is defined %}
      SET_KINEMATIC_POSITION Z=1.9 # Set Z position
      G1 Z4 F600 # Lower Z by 4 to prevent dragging the nozzle
      SET_TMC_CURRENT STEPPER=stepper_x CURRENT={HOME_CUR * 0.7}
      G28 X
      SET_TMC_CURRENT STEPPER=stepper_x CURRENT={HOME_CUR}
      BEEP I=1 DUR=100
      G1 X10 F1200
```

```
{% endif %}


{% if params.Y is defined %}
    SET_KINEMATIC_POSITION Z=1.9 # Set Z position
    G1 Z4 F600 # Lower Z by 4 to prevent dragging the nozzle
    SET_TMC_CURRENT STEPPER=stepper_y CURRENT={HOME_CUR * 0.9}
    G28 Y
    SET_TMC_CURRENT STEPPER=stepper_y CURRENT={HOME_CUR}
    BEEP I=1 DUR=100
   G1 Y10 F1200
{% endif %}


{% if params.Z is defined %}
    G28 x
    G28 Y
    G28 X
    G1 X150 Y150 F7800
    SET_KINEMATIC_POSITION Z={printer.toolhead.axis_maximum.z-30}
    probe
    SET_KINEMATIC_POSITION Z=-0.1
    G1 Z30 F480
{% endif %}


{% if params.X is undefined %}
```

```
{% if params.Y is undefined %}
{% if params.Z is undefined %}
  SET_KINEMATIC_POSITION X=0
  SET_KINEMATIC_POSITION Y=0
  SET_KINEMATIC_POSITION Z={printer.toolhead.axis_maximum.z-30}
  G91
  G1 Z7 F600
  G1 X5 F2400
  G1 Y5 F2400
  G4 P2000

 SET_TMC_CURRENT STEPPER=stepper_x CURRENT={HOME_CUR * 0.8}
  G28 X
  SET_TMC_CURRENT STEPPER=stepper_x CURRENT={HOME_CUR}
  BEEP I=1 DUR=100
  G1 X45 F1200

  SET_TMC_CURRENT STEPPER=stepper_y CURRENT={HOME_CUR * 0.9}
  G28 Y
  SET_TMC_CURRENT STEPPER=stepper_y CURRENT={HOME_CUR}
  BEEP I=1 DUR=100
  G1 Y10 F1200

  SET_TMC_CURRENT STEPPER=stepper_x CURRENT={HOME_CUR * 0.8}
```

```
    G28 X
    SET_TMC_CURRENT STEPPER=stepper_x CURRENT={HOME_CUR}
    BEEP I=1 DUR=100
    G1 X10 F1200


    SET_KINEMATIC_POSITION Z={printer.toolhead.axis_maximum.z-10}


    G90
    G1 X150 Y150 F7800
    G91
    G28 Z
    G1 Z30  F600
{% endif %}
{% endif %}
{% endif %}
SET_TMC_CURRENT STEPPER=stepper_x CURRENT={RUN_CUR}
SET_TMC_CURRENT STEPPER=stepper_y CURRENT={RUN_CUR}
M204 S10000
G90
```

Найдите и замените `[gcode_macro G29]` на это:

```
[gcode_macro G29]
  variable_k:1
```

```
gcode:
  M141 S0
  BED_MESH_CLEAR
  SET_GCODE_OFFSET Z=0
  {% if "xy" not in printer.toolhead.homed_axes %}
    G28 Y                         # Home Y axis
    G0 Y20 F1200                  # Move Y axis away from Y end-stop
    G28 X                         # Home X axis
  {% endif %}
  M109 S145                       # Set nozzle to 145 so any remaining filament
stuck to nozzle is softened
  G28 Z                           # Home Z
  CARTOGRAPHER_CALIBRATE SPEED=2        # Re-Calibrate incase build plate
changes
  Z_TILT_ADJUST                   # Ensure bed is level
  G28 Z                           # Re-home Z again now that the bed is level
  M109 S0                         # Turn off hotend
  {% if k|int==1 %}
    BED_MESH_CALIBRATE RUNS=2 PROFILE=kamp
    BED_MESH_PROFILE LOAD=kamp
    SAVE_VARIABLE VARIABLE=profile_name VALUE='"kamp"'
  {% else %}
    BED_MESH_CALIBRATE RUNS=2 PROFILE=default
    BED_MESH_PROFILE LOAD=default
    SAVE_VARIABLE VARIABLE=profile_name VALUE='"default"'
```

```
    {% endif %}

    CARTOGRAPHER_TOUCH SPEED=2 FUZZY=10
```

Наконец, удалите или закомментируйте следующие строки, чтобы предотвратить сохранение старой переменной Z_offset.

```
#PRINT_END

#[gcode_macro PRINT_END]

#gcode:

#{% if printer.gcode_move.homing_origin.z < 0.5 %}

#SAVE_VARIABLE VARIABLE=z_offset
VALUE={printer.gcode_move.homing_origin.z}

#{% endif %}

#[gcode_macro CANCEL_PRINT]

#save_zoffset
```

5.3 Изменения в saved_variables.cfg

Внутри `saved_variables.cfg` нам нужно установить `z_offset = 0.0`, так как этим теперь занимается Cartographer.

6. Создайте новый файл в папке конфигурации с именем carto.cfg .

image.png

Скопируйте в новый файл конфигурацию. Сохраните и закройте. Данные в конфигурации (кроме серийника Cartographer) подходят если вы используете крепление из начала статьи

> [mcu scanner]
>
> serial: /dev/serial/by-id/usb-C_614e_xxxxxxxxxxxx
>
> # adjust to suit your scanner, if using usb change to serial
>
> [scanner]
>
> mcu: scanner
>
> # Offsets are measured from the centre of your coil, to the tip of your nozzle
>
> # on a level axis. It is vital that this is accurate.
>
> x_offset: 0 # Adjust this value based on your mount
>
> # adjust for your cartographers offset from nozzle to middle of coil
>
> y_offset: -18.8 # Adjust this value based on your mount
>
> # adjust for your cartographers offset from nozzle to middle of coil
>
> backlash_comp: 0.00499
>
> # Backlash compensation distance for removing Z backlash before measuring
>
> # the sensor response.
>
> sensor: cartographer
>
> # this must be set as cartographer unless using IDM etc.
>
> sensor_alt: carto
>
> # alternate name to call commands. CARTO_TOUCH etc
>
> mesh_runs: 2
>
> # Number of passes to make during mesh scan.

```
[temperature_sensor Cartographer_MCU]

sensor_type: temperature_mcu

sensor_mcu: scanner

min_temp: 0

max_temp: 105



#These values work for Spooknik's side mount

[z_tilt]

z_positions:

  -17.5,138.5

  335.7,138.5



points:

  50, 170.8   # Based on Stew's mount, adjust accordingly

  255, 170.8  # Based on Stew's mount, adjust accordingly



speed: 150

horizontal_move_z: 5

retries: 2

retry_tolerance: 0.05
```

Откройте `printer.cfg` и вверху под последним include напишите `[include carto.cfg]` . Сохраните и перезапустите.

В этом файле мы храним все изменения в printer.cfg, которые нужны Carto для работы. Вам нужно изменить несколько значений в этом файле, чтобы ваш carto работал для вашей настройки.

В carto.cfg нужно прописать серийный номер вашего Cartographer

> `serial: /dev/serial/by-id/[ВАШ СЕРИЙНИК]`

Найти серийник можно запустив в SSH.

> `ls /dev/serial/by-id/*`

Должно получиться что-то вроде:

> `/dev/serial/by-id/usb-Cartographer_614e_XXXXXXXXXXXXXXXXXXXXXXXXXX`

`Скопируйте все что идет после **by-id/** и вставьте в` carto.cfg в разделе [mcu scanner]

X Y Offset

Эти данные должен предоставить автор используемого вами крепления. Расстояние по осям X и Y от центра катушки до центра сопла.

```
x_offset: 0 # Based on Stew's mount, adjust accordingly
```

```
y_offset: -18.8 # Based on Stew's mount, adjust accordingly
```

[z_tilt]

В зависимости от смещения X и Y вашего крепления относительно сопла, вам необходимо обновить точки измерения z, чтобы они измеряли над ходовыми винтами.

Обычно вам необходимо применить смещение Z вашего крепления зонда к точкам измерения Z.

Вы также можете вручную переместить головку инструмента в Fluidd и записать координаты X и Y. Для получения дополнительной информации см. документы klipper

https://www.klipper3d.org/Config_Reference.html?h=z_tilt#z_tilt

Приведенные ниже значения верны, если вы используете крепление из начала статьи, и приведены просто в качестве примера.

```
points:
```

```
    50, 170.8   # Based on Stew's mount, adjust accordingly
```

```
    255, 170.8  # Based on Stew's mount, adjust accordingly
```

[bed_mesh]

В **printer.cfg** `добавим новую конфигурацию сетки кровати`

> [bed_mesh]
>
> zero_reference_position: 150, 150
>
> speed: 250
>
> horizontal_move_z: 5
>
> mesh_min: 22,30   # Based on Stew's mount, adjust accordingly
>
> mesh_max: 280,280 # Based on Stew's mount, adjust accordingly
>
> probe_count: 25, 25
>
> algorithm: bicubic

Подобно `z_tilt` вам нужно настроить `mesh_min` и `mesh_max`, чтобы они соответствовали смещению вашего зонда. Вам нужно изменить смещение X и Y в соответствии со смещением вашего крепления, чтобы зонд всегда сканировал над кроватью

Приведенные ниже значения верны, если вы используете крепление из начала статьи, и приведены просто в качестве примера.

```
mesh_min: 22,22   # Based on Stew's mount, adjust accordingly
```

```
mesh_max: 283,283 # Based on Stew's mount, adjust accordingly
```

Дополнительные макросы Выравнивания стола

Благодаря датчику Cartographer, который теперь обеспечивает точные измерения *Z-Offset* его можно использовать для значительного упрощения задачи выравнивания стола с помощью 4 ручек под печатной платформой.

Добавьте следующие макросы в конец вашего файла `gcode_macro.cfg`:

> [gcode_macro SFL]
>
> description: Get zoffset at front-left bed adjustment screw position
>
> gcode:
>
> G1 X{25 - printer.configfile.settings.scanner.x_offset} Y{21 - printer.configfile.settings.scanner.y_offset} F6000
>
> PROBE
>
> [gcode_macro SFR]
>
> description: Get zoffset at front-right bed adjustment screw position
>
> gcode:
>
> G1 X{285 - printer.configfile.settings.scanner.x_offset} Y{21 - printer.configfile.settings.scanner.y_offset} F6000
>
> PROBE
>
> [gcode_macro SBL]
>
> description: Get zoffset at back-left bed adjustment screw position
>
> gcode:
>
> G1 X{25 - printer.configfile.settings.scanner.x_offset} Y{281 - printer.configfile.settings.scanner.y_offset} F6000

```
  PROBE


[gcode_macro SBR]

description: Get zoffset at back-right bed adjustment screw position

gcode:

  G1 X{285 - printer.configfile.settings.scanner.x_offset} Y{281 -
printer.configfile.settings.scanner.y_offset} F6000

  PROBE
```

Каждый из макросов выше будет располагать датчик над ручками, чтобы вы могли быстро отрегулировать и выполнить повторное измерение.

Чтобы использовать эти макросы,сначала очистите сетку стола, припаркуйте оси и повторно откалибруйте датчик, вызвав следующие макросы:

```
M84

BED_MESH_CLEAR

SET_GCODE_OFFSET Z=0

G28

Z_TILT_ADJUST

G28
```

затем вызовите макросы, перечисленные выше, и посмотрите на последнюю строку (например, `// Result is z=1.948191`). Это информирует вас о том, насколько далеко находится печатная платформа от зонда. Вы можете отрегулировать ручку под платформой и снова вызвать тот же макрос, чтобы получить новое значение. Это можно повторить для каждой из 4 точек стола, пока все не будут равны в пределах ~0,02 мм. Будет сложно получить большую точность, чем эта. По умолчанию зонд Cartographer устанавливает голову на высоту 2,00 мм от стола после вызова **`G28`**, поэтому мы стремимся, чтобы все 4 положения винтов были большее **`z=1.98`** и меньшее, чем **`z=2.02`**

`Выравниванием стола стоит заняться после калибровки` Cartographer.

Сохраняем все изменения.

7.1. Калибровка Cartographer.

(Далее все команды вводятся в консоли Fluidd)

Если вы хотите использовать Cartographer для автоматического определения Z-*Offset*, вам необходимо будет следовать всем дополнительным инструкциям `TOUCH` .

Сначала убедитесь, что вы находитесь в режиме `TOUCH`.

В консоли **Fluidd** введите:

> PROBE_SWITCH MODE=touch

Далее обязательно сохраните, введя:

> SAVE_CONFIG

Припаркуйте оси X и Y введя в консоли:

> G28 X Y

Передвиньте сопло в центр стола введя:

> G0 X150 Y150

7.2. Запустите процесс калибровки

В консоли **Fluidd** введите:

> CARTOGRAPHER_CALIBRATE METHOD=manual

Вы можете использовать веб-интерфейс для регулировки высоты сопла от кровати. Используйте лист бумаги или щуп для измерения Z-*Offset*. После завершения удалите бумагу/щуп и примите положение введя:

> ACCEPT

Сохраните результаты в файле конфигурации введя.

> SAVE_CONFIG

Припаркуйте оси введя:

> G28

Теперь можно проверить точность катушки сканера введя:

> PROBE_ACCURACY

Сопло не будет КАСАТЬСЯ стола. Будет использоваться сканер для измерения точности зондовой катушки.

Вы также можете измерить люфт оси Z введя:

> CARTOGRAPHER_ESTIMATE_BACKLASH

Чтобы использовать оценку люфта, которая получилась. Возьмите результат, начинающийся начиная с

`Median distance moving up`

В этой строке будет измерение под названием `"delta"`. Запишите значение в файл `carto.cfg`. Найдите раздел конфигурации, обозначенный:

>

> `backlash_comp: 0.00499`

Теперь выполняем выравнивание стола макросами из инструкции выше: `SFL,SFR,SBL,SBR.`

`После выравнивания стола винтами вы можем запустить калибровку сетки стола введя:`

> BED_MESH_CALIBRATE

7.3. Настройка касания сопла

Выполнить возврат в исходное положение введя:

> G28

Далее необходимо измерить наклон оси Z введя:

> Z_TILT_ADJUST

Как только измерение будет сделано, сделайте еще одну парковку осей:

> G28

Теперь можем запустить пороговое сканирование. Cartographer определит, какое усилие требуется чтобы постоянно касаться стола.

Перед выполнением порогового сканирования убедитесь, что выполняются следующие условия.

- Ваше сопло чистое
- Ваша постель чистая, в месте, которое вы проверяете, нет никаких вмятин или пузырей.
- Ваш зонд находится на расстоянии 2,6–3 мм от сопла.
- Ваша кинематика жесткая.

Введите:

> CARTOGRAPHER_THRESHOLD_SCAN

Это запустит процесс касания соплом стола, переместит голову в исходное положение, а затем опустит ее, пока она не коснется стола. Ничего страшного, если сначала сопло вообще не коснется стола, это совершенно нормально. В конце концов сопло начнет касаться.

Однако если процесс закончился и сопло не коснулось стола, начните процесс заново или настройте параметры следующим образом

Пример:

> CARTOGRAPHER_THRESHOLD_SCAN MIN=500

где MIN = найденное пороговое значение ложного срабатывания.

Как только найдется оптимальный порог и вы увидите, как сопло касается стола, процесс остановится.

Вы можете остановиться на этом, но для большинства тонкая настройка может быть полезной.

Повторно запустите команду, установив значение MIN немного выше (+250) предыдущего оптимального результата и дайте ей снова запуститься.

Теперь выполните калибровку касания с новым порогом введя:

> CARTOGRAPHER_CALIBRATE

Если все выполнено правильно тест касания должен пройти успешно и теперь вы можете завершить его сохранив эти переменные в своей конфигурации введя:

> SAVE_CONFIG

Калибровка и настройка CARTOGRAPHER закончена.

Так как мы добавили на голову вес в виде нового крепления и датчика, рекомендуется заново произвести калибровку input shaper

Статью перевел специально для WIKI: @WoolfRS

Revision #2
Created 2026-10-08 10:26:02 UTC by Volvik88
Updated 2026-10-09 14:01:02 UTC by Volvik88