

Halcyon

Инструкция.



Оглавление

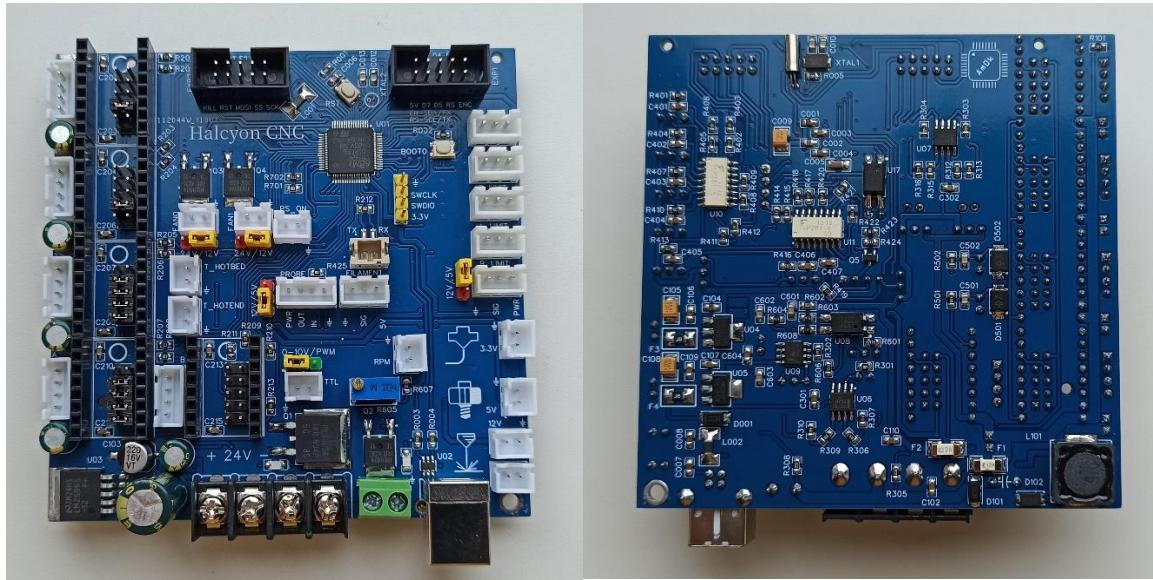
Общая информация.....	3
Внешний вид.....	4
Работа с прошивкой Marlin.....	5
Настройка и загрузка прошивки.....	5
Добавление своих терморезисторов в Marlin.....	7
Работа с прошивкой GRBL.....	8
Подключение периферийных устройств.....	12
Выходы HOTBED, TTL, RPM.....	13
Подключение концевых выключателей.....	14
Выбор режима работы и микрошага драйверов шаговых двигателей.....	15
Подключение датчика BL-TOUCH.....	16
Подключение разъемов дополнительной периферии EXP1 и EXP2.....	17
Дополнительная информация.....	18
Размеры.....	18
Pinout.....	19

Общая информация.

Плата построена на базе современного и дешевого 32-битного МК STM32F401RCT6 / RET6.

- Питание платы осуществляется от блока питания 24 В, никаких дополнительных линий питания не требуется.
- Возможность подключения до 5 независимых осей. Реализовано управление драйверами ШД через UART.
- Питание нагревателя экструдера и шаговых двигателей осуществляется напряжением первичного питания (24 В).
- Для подключения нагревателя стола или шпинделя на плате необходимо установить перемычку, выбрав один из вариантов: 0-10V для подключения шпинделей, управляемых внешним драйвером с возможностью задавать обороты аналоговым сигналом RPM, снимаемым с соответствующего разъема на плате, либо PWM, если необходимо управление сигналом TTL с соответствующего разъема (для лазеров или SSR, через которых подключен нагреватель стола). Так же при установленной перемычке PWM возможно подключение через встроенный силовой мосфет (для 24В нагревателей или коллекторных шпинделей с током потребления до 12А).
- 2 управляемых вентилятора с возможностью выбора напряжения питания для каждого из них.
- 5 независимых концевых выключателей, имеющих опоразвязку. Напряжение питания концевых выключателей может быть выбрано. Плата рассчитана на работу с концевыми выключателями, в которых общий (COM) сигнальный контакт нормально-замкнут на землю и при срабатывании коммутируется с напряжением питания.
- Разъем `PROBE` для подключения датчиков автоуровня / длины инструмента (с опоразвязкой входного и выходного сигналов).
- Разъем `FILAMENT` для подключения датчика филамента (с опоразвязкой).
- Возможность реализовать автоотключение внешним реле через сигнал PS_ON (опоразвязан, с открытым коллектором).
- Изменена входная цепочка измерения температуры нагревателя экструдера. Подтягивающий резистор установлен с сопротивлением 1kΩ, что обеспечивает большее количество отсчетов АЦП в диапазоне температур 80-300 С, а значит и большую точность поддержания заданной температуры. К сожалению, это снижает количество отсчетов АЦП при температурах, близких к комнатным, однако, они не являются рабочими. Входы подключения термисторов защищены супрессорами.
- На дополнительные разъемы EXP1 и EXP2 выведены интерфейсы SPI, UART, I2C для подключения внешних устройств. Плата протестирована с модулями дисплеев RepRap Discount Smart Controller и MKS MINI12864 V3.
- Дополнительное питания 3.3 В, 5 В, 12В выведено с платы на соответствующие разъемы.
- USB интерфейс защищен от электростатических разрядов.
- Возможность прошивки платы через ST-LINK, либо по DFU протоколу через USB, переведя плату в режим DFU (для этого необходимо перезагрузить плату с нажатой кнопкой BOOT0).

Внешний вид.



Работа с прошивкой Marlin.

Настройка и загрузка прошивки.

- Добавляем файл определения пинов:

```
.. \ Marlin\src\pins\stm32f4\pins_HALCYON_V1.h
```

- Добавляем в файл

```
.. \ Marlin\src\pins\pins.h:
```

```
#elif MB(HALCYON_V1)
    #include "stm32f4/pins_HALCYON_V1.h"           // STM32F4
env:Halcyon_v1_RC env:Halcyon_v1_RC_dfu env:Halcyon_v1_RC_stlink env:Halcyon_v1_RE
env:Halcyon_v1_RE_dfu env:Halcyon_v1_RE_stlink
```

Важно: в закомментированной части строки есть информация о необходимом окружении, она парсится скриптом во время подготовки к компиляции, поэтому должна соответствовать имени вашего окружения на следующем шаге.

- Добавляем в файл .. \Marlin\src\core\boards.h:

```
#define BOARD_HALCYON_V1           5245 // HALCYON BOARD
```

Номер после названия платы может быть любым (при выходе свежих версий система нумерации периодически изменяется, а уже существующие номера сдвигаются), главное чтобы он был уникальным.

- Добавляем/редактируем в файле .. \Marlin\Configuration.h следующую информацию:

```
#ifndef MOTHERBOARD
#define MOTHERBOARD BOARD_HALCYON_V1
#endif

#define SERIAL_PORT          -1
#define BAUDRATE             250000
```

- Добавляем информацию об окружении для платы в файл .. \ini\stm32f4.ini:

```
#
# Halcyon_v1 board
# STM32F401RCT6 ARM Cortex-M4
# use BOOTLOADER_HALCYON_V1_RC_SPI_SD.bin
#
[env:Halcyon_v1_RC]
extends                  = stm32_variant
platform                 = ststm32
platform_packages         = framework-arduinostm32@~4.20600.231001
                           framework-cmsis@~2.50700.210515
                           toolchain-gccarmnoneabi@1.100301.220327
board                    = genericSTM32F401RC
board_build.offset        = 0x8000
board_upload.offset_address = 0x08008000
build_flags               = ${stm32_variant.build_flags}
${stm32f4_I2C1.build_flags}
                           -Os -DHAL_PCD_MODULE_ENABLED
                           -DHAL_UART_MODULE_ENABLED
                           -DPIN_WIRE_SCL=PB6 -DPIN_WIRE_SDA=PB7
                           -DSERIAL_RX_BUFFER_SIZE=1024 -DSERIAL_TX_BUFFER_SIZE=1024
                           -DTIMER_SERVO=TIM2
```

```

monitor_speed           = DSTEP_TIMER_IRQ_PRIO=0
                        = 250000

[env:Halcyon_v1_RC_dfu]
extends                 = env:Halcyon_v1_RC
upload_protocol         = dfu

[env:Halcyon_v1_RC_stlink]
extends                 = env:Halcyon_v1_RC
upload_protocol         = stlink

#
# Halcyon_v1 board
# STM32F401RET6 ARM Cortex-M4
# use BOOTLOADER_HALCYON_V1_RC_SPI_SD.bin
#
[env:Halcyon_v1_RE]
extends                 = stm32_variant
platform                = ststm32
platform_packages        = framework-arduinooststm32@~4.20600.231001
                           framework-cmsis@~2.50700.210515
                           toolchain-gccarmnoneeabi@1.100301.220327
board                   = genericSTM32F401RE
board_build.offset      = 0x8000
board_upload.offset_address = 0x08008000
build_flags              = ${stm32_variant.build_flags}
${stm32f4_I2C1.build_flags}
                           -Os -DHAL_PCD_MODULE_ENABLED
                           -DHAL_UART_MODULE_ENABLED
                           -DPIN_WIRE_SCL=PB6 -DPIN_WIRE_SDA=PB7
                           -DSERIAL_RX_BUFFER_SIZE=1024 -DSERIAL_TX_BUFFER_SIZE=1024
                           -DTIMER_SERVO=TIM2
                           -DSTEP_TIMER_IRQ_PRIO=0
monitor_speed           = 250000

[env:Halcyon_v1_RE_dfu]
extends                 = env:Halcyon_v1_RE
upload_protocol         = dfu

[env:Halcyon_v1_RE_stlink]
extends                 = env:Halcyon_v1_RE
upload_protocol         = stlink

```

6. Загрузите *BOOTLOADER_HALCYON_V1_RC_SPI_SD.bin* (для STM32F401RCT6 контроллера) / *BOOTLOADER_HALCYON_V1_RC_SPI_SD.bin* (для STM32F401RET6 контроллера), если он еще не был загружен на вашу плату, по USB (для этого необходимо перезагрузить плату кнопкой RST с зажатой кнопкой BOOT0), либо с помощью ST-LINK через специальный разъем программирования на плате.
7. Скомпилируйте прошивку используя соответствующее вашему микроконтроллеру окружение: [env:Halcyon_v1_RC] ([env:Halcyon_v1_RE]). Теперь вы можете обновлять прошивку вашей платы просто поместив скомпилированный файл прошивки **firmware.bin** на SD карту и подав после этого питание на принтер. В течении ~30 секунд произойдет загрузка прошивки, в случае успеха файл на SD карте будет переименован в **firmware.cur**. Вы также можете прошить плату, выбрав окружение [env:Halcyon_v1_RC_dfu] ([env:Halcyon_v1_RE_dfu]) для прошивки по USB, либо [env:Halcyon_v1_RC_stlink] ([env:Halcyon_v1_RE_stlink]) для прошивки с помощью ST-LINK через специальный разъем программирования на плате.

Добавление своих терморезисторов в Marlin.

1. Создаем header-файл с калибровкой терморезистора и именем
.. \Marlin\src\module\thermistor\thermistor_59.h:

```
#pragma once

constexpr temp_entry_t temptable_59[] PROGMEM = {
    { OV(91), 300 },
    { OV(106), 290 },
    { OV(121), 280 },
    { OV(140), 270 },
    { OV(161), 260 },
    { OV(186), 250 },
    { OV(217), 240 },
    { OV(248), 230 },
    { OV(287), 220 },
    { OV(330), 210 },
    { OV(374), 200 },
    { OV(426), 190 },
    { OV(478), 180 },
    { OV(536), 170 },
    { OV(594), 160 },
    { OV(654), 150 },
    { OV(709), 140 },
    { OV(760), 130 },
    { OV(807), 120 },
    { OV(850), 110 },
    { OV(887), 100 },
    { OV(919), 90 },
    { OV(944), 80 },
    { OV(964), 70 },
    { OV(980), 60 },
    { OV(993), 50 },
    { OV(1001), 40 },
    { OV(1006), 30 },
    { OV(1007), 25 },
    { OV(1017), 0 }
};
```

2. В файл записываем свою таблицу калибровки, показания АЦП (чтобы их видеть в консоли, необходимо раскомментировать параметр **#define SHOW_TEMP_ADC_VALUES** в файле .. \Marlin\Configuration_adv.h) и показания реальной температуры, полученной с помощью своей термопары или иным заведомо точным способом. Значения АЦП записываются в 10-битном виде, поэтому если у вас АЦП имеет 12 бит, то значения стоит делить на 4 (2 в степени битность вашего АЦП минус 10). Чем чаще шаг по температуре, тем лучше, но чаще 10 смысла делать нет, верхний предел записываем тот, который нам необходим (очевидно, нет смысла калибровать стол до 270 градусов).
3. В файл .. \Marlin\src\module\thermistor\thermistors.h добавляем строки:

```
#if ANY_THERMISTOR_IS(59)
    #include "thermistor_59.h"
#endif
```

4. В файле .. \Marlin\Configuration.h определяем как используемый наш резистор:

```
#define TEMP_SENSOR_0 59
```

Работа с прошивкой GRBL.

Плата официально поддерживается и добавлена в оригинальную прошивку, поэтому рекомендуется использовать последнюю стабильную версию оригинальной прошивки [GRBL HAL](#).

Ознакомьтесь со следующей [страницей](#) перед первым запуском GRBL чтобы избежать непонятных ситуаций.

Если хотите облегчить редактирование настроек по умолчанию, выполните следующие два шага:

1. Добавляем в файл `.\grbl\config.h` после строки `#define _GRBL_CONFIG_H_`:

```
#include "my_cfg.h"
```

2. Создаем файл `.\grbl\my_cfg.h` и внесите необходимые изменения:

```
#define BUILD_INFO                                "my_cfg.h"
#define N_AXIS                                     4
#define N_SPINDLE                                  1
#define N_SYS_SPINDLE                            1
#define MASLOW_ROUTER                           Off
#define WALL_PLOTTER                            Off
#define DELTA_ROBOT                             Off
#define POLAR_ROBOT                            Off
#define COREXY                                 Off

#define CHECK_MODE_DELAY                         0
#define DEBOUNCE_DELAY                           40

#define ENABLE_ACCELERATION_PROFILES           Off
#define ENABLE_JERK_ACCELERATION                Off

#define ACCELERATION_TICKS_PER_SECOND          100
#define REPORT_ECHO_LINE_RECEIVED              Off
#define TOOL_LENGTH_OFFSET_AXIS                  -1
#define MINIMUM_JUNCTION_SPEED                 0.0f
#define MINIMUM_FEED_RATE                      1.0f
#define N_ARC_CORRECTION                        12
#define ARC_ANGULAR_TRAVEL_EPSILON            5E-7f
#define BEZIER_MIN_STEP                          0.002f
#define BEZIER_MAX_STEP                          0.1f
#define BEZIER_SIGMA                            0.1f
#define DWELL_TIME_STEP                         50
#define SEGMENT_BUFFER_SIZE                     10
#define SET_CHECK_MODE_PROBE_TO_START           Off
#define HARD_LIMIT_FORCE_STATE_CHECK           Off
#define HOMING_AXIS_SEARCH_SCALAR              1.5f
#define HOMING_AXIS_LOCATE_SCALAR               10.0f

#define ENABLE_RESTORE_NV_S_WIPE_ALL             On
#define ENABLE_RESTORE_NV_DEFAULT_SETTINGS      On
#define ENABLE_RESTORE_NV_CLEAR_PARAMETERS      On
#define ENABLE_RESTORE_NV_DRIVER_PARAMETERS     On
#define SETTINGS_RESTORE_DEFAULTS               On
#define SETTINGS_RESTORE_PARAMETERS              On
#define SETTINGS_RESTORE_STARTUP_LINES          On
#define SETTINGS_RESTORE_BUILD_INFO              On
#define SETTINGS_RESTORE_DRIVER_PARAMETERS      On
#define DISABLE_BUILD_INFO_WRITE_COMMAND        Off

#define SLEEP_DURATION                           5.0f
#define NVSDATA_BUFFER_ENABLE                   On

#define TOOLSETTER_RADIUS                        5.0f
#define ENABLE_BACKLASH_COMPENSATION             Off
#define N_TOOLS                                  0
#define SPINDLE_SYNC_ENABLE                      Off
#define NGC_EXPRESSIONS_ENABLE                  Off
#define NGC_PARAMETERS_ENABLE                   On
```

#define LATHE_UVW_OPTION	Off	
#define DEFAULT_STEP_PULSE_MICROSECONDS	10.0f	//\$0
#define DEFAULT_STEPPER_IDLE_LOCK_TIME	255	//\$1
#define DEFAULT_STEP_SIGNALS_INVERT_MASK	0b00000000	//\$2
#define DEFAULT_DIR_SIGNALS_INVERT_MASK	0b00000001	//\$3
#define DEFAULT_ENABLE_SIGNALS_INVERT_MASK	0b00001111	//\$4
#define DEFAULT_LIMIT_SIGNALS_INVERT_MASK	0b00000000	//\$5
#define DEFAULT_PROBE_SIGNAL_INVERT	On	//\$6
#define DEFAULT_TOOLSETTER_SIGNAL_INVERT	Off	//\$6
#define DEFAULT_GANGED_DIRECTION_INVERT_MASK	0b00000000	//\$8
#define DEFAULT_SPINDLE_ENABLE_OFF_WITH_ZERO_SPEED	On	//\$9
#define DEFAULT_PWM_SPINDLE_DISABLE_LASER_MODE	Off	//\$9
#define DEFAULT_REPORT_MACHINE_POSITION	On	//\$10(bit 0)
#define DEFAULT_REPORT_BUFFER_STATE	On	//\$10(bit 1)
#define DEFAULT_REPORT_LINE_NUMBERS	On	//\$10(bit 2)
#define DEFAULT_REPORT_CURRENT_FEED_SPEED	On	//\$10(bit 3)
#define DEFAULT_REPORT_PIN_STATE	On	//\$10(bit 4)
#define DEFAULT_REPORT_WORK_COORD_OFFSET	On	//\$10(bit 5)
#define DEFAULT_REPORT_OVERRIDES	On	//\$10(bit 6)
#define DEFAULT_REPORT_PROBE_COORDINATES	On	//\$10(bit 7)
#define DEFAULT_REPORT_SYNC_ON_WCO_CHANGE	On	//\$10(bit 8)
#define DEFAULT_REPORT_PARSER_STATE	Off	//\$10(bit 9)
#define DEFAULT_REPORT_ALARM_SUBSTATE	Off	//\$10(bit 10)
#define DEFAULT_REPORT_RUN_SUBSTATE	Off	//\$10(bit 11)
#define DEFAULT_JUNCTION_DEVIATION	0.01f	//\$11
#define DEFAULT_ARC_TOLERANCE	0.002f	//\$12
#define DEFAULT_REPORT_INCHES	Off	//\$13
#define DEFAULT_CONTROL_SIGNALS_INVERT_MASK	0b00000000	//\$14
#define DEFAULT_INVERT_COOLANT_FLOOD_PIN	Off	//\$15
#define DEFAULT_INVERT_COOLANT_MIST_PIN	Off	//\$15
#define DEFAULT_INVERT_SPINDLE_ENABLE_PIN	Off	//\$16
#define DEFAULT_INVERT_SPINDLE_CCW_PIN	Off	//\$16
#define DEFAULT_INVERT_SPINDLE_PWM_PIN	Off	//\$16
#define DEFAULT_DISABLE_CONTROL_PINS_PULL_UP_MASK	0b00000000	//\$17
#define DEFAULT_LIMIT_SIGNALS_PULLUP_DISABLE_MASK	0b00000000	//\$18
#define DEFAULT_PROBE_SIGNAL_DISABLE_PULLUP	Off	//\$19
#define DEFAULT_TOOLSETTER_SIGNAL_DISABLE_PULLUP	Off	//\$19
#define DEFAULT_SOFT_LIMIT_ENABLE	On	//\$20
#define DEFAULT_HARD_LIMIT_ENABLE	On	//\$21
#define DEFAULT_CHECK_LIMITS_AT_INIT	On	
#define DEFAULT_HARD_LIMITS_DISABLE_FOR_ROTARY	On	
#define DEFAULT_HOMING_ENABLE	On	//\$22(bit 0)
#define DEFAULT_HOMING_SINGLE_AXIS_COMMANDS	Off	//\$22(bit 1)
#define DEFAULT_HOMING_INIT_LOCK	On	//\$22(bit 2)
#define DEFAULT_HOMING_FORCE_SET_ORIGIN	Off	//\$22(bit 3)
#define DEFAULT_LIMITS_TWO_SWITCHES_ON_AXES	Off	//\$22(bit 4)
#define DEFAULT_HOMING_ALLOW_MANUAL	On	//\$22(bit 5)
#define DEFAULT_HOMING_OVERRIDE_LOCKS	On	//\$22(bit 6)
#define DEFAULT_HOMING_USE_LIMIT_SWITCHES	Off	//\$22(bit 8)
#define DEFAULT_RUN_STARTUP_SCRIPTS_ONLY_ON_HOMED	Off	//\$22(bit 10)
#define DEFAULT_HOMING_DIR_MASK	0b00000111	//\$23
#define DEFAULT_HOMING_FEED_RATE	25.0f	//\$24
#define DEFAULT_HOMING_SEEK_RATE	300.0f	//\$25
#define DEFAULT_HOMING_DEBOUNCE_DELAY	200	//\$26
#define DEFAULT_HOMING_PULLOFF	1.0f	//\$27
#define DEFAULT_G73_RETRACT	0.1f	//\$28
#define DEFAULT_STEP_PULSE_DELAY	0.0f	//\$29
#define DEFAULT_SPINDLE_RPM_MAX	12000.0f	//\$30
#define DEFAULT_SPINDLE_RPM_MIN	3000.0f	//\$31
#define DEFAULT_LASER_MODE	Off	//\$32
#define DEFAULT_LATHE_MODE	Off	//\$32
#define DEFAULT_SPINDLE_PWM_FREQ	5000	//\$33
#define DEFAULT_SPINDLE_PWM_OFF_VALUE	0.0f	//\$34
#define DEFAULT_SPINDLE_PWM_MIN_VALUE	0.0f	//\$35
#define DEFAULT_SPINDLE_PWM_MAX_VALUE	100.0f	//\$36
#define DEFAULT_STEPPER_DEENERGIZE_MASK	0b00000000	//\$37
#define DEFAULT_SPINDLE_PPR	0	//\$38
#define DEFAULT_LEGACY_RTCOMMANDS	On	//\$39
#define DEFAULT_JOG_LIMIT_ENABLE	Off	//\$40
#define DEFAULT_PARKING_ENABLE	Off	//\$41(bit 0)
#define DEFAULT_DEACTIVATE_PARKING_UPON_INIT	Off	//\$41(bit 1)
#define DEFAULT_ENABLE_PARKING_OVERRIDE_CONTROL	Off	//\$41(bit 2)
#define DEFAULT_PARKING_AXIS	0b00000100	//\$42
#define DEFAULT_N_HOMING_LOCATE_CYCLE	1	//\$43
#define DEFAULT_HOMING_CYCLE_0	0b00000100	//\$44
#define DEFAULT_HOMING_CYCLE_1	0b00000111	//\$45
#define DEFAULT_HOMING_CYCLE_2	0	//\$46

#define DEFAULT_HOME_OFFSET_X	0	///\$47
#define DEFAULT_HOME_OFFSET_Y	0	///\$48
#define DEFAULT_HOME_OFFSET_Z	0	///\$49
#define DEFAULT_PARKING_PULLOUT_INCREMENT	5.0f	///\$56
#define DEFAULT_PARKING_PULLOUT_RATE	100.0f	///\$57
#define DEFAULT_PARKING_TARGET	-5.0f	///\$58
#define DEFAULT_PARKING_RATE	500.0f	///\$59
#define DEFAULT_RESET_OVERRIDES	Off	///\$60
#define DEFAULT_DOOR_IGNORE_WHEN_IDLE	Off	///\$61
#define DEFAULT_SLEEP_ENABLE	Off	///\$62
#define DEFAULT_ENABLE_LASER_DURING_HOLD	On	///\$63
#define DEFAULT_RESTORE_AFTER_FEED_HOLD	On	///\$63
#define DEFAULT_FORCE_INITIALIZATION_ALARM	Off	///\$64
#define DEFAULT_ALLOW_FEED_OVERRIDE_DURING_PROBE_CYCLES	Off	///\$65
#define DEFAULT_SOFT_LIMIT_PROBE_CYCLES	Off	///\$65
#define DEFAULT_X_STEPS_PER_MM	6400.0f	///\$100
#define DEFAULT_Y_STEPS_PER_MM	6400.0f	///\$101
#define DEFAULT_Z_STEPS_PER_MM	1600.0f	///\$102
#define DEFAULT_A_STEPS_PER_MM	8.88888889f	///\$103
#define DEFAULT_X_MAX_RATE	600.0f	///\$110
#define DEFAULT_Y_MAX_RATE	600.0f	///\$111
#define DEFAULT_Z_MAX_RATE	400.0f	///\$112
#define DEFAULT_A_MAX_RATE	(1.0f * 360.0f * 60.0f)	///\$113
#define DEFAULT_X_ACCELERATION	12.0f	///\$120
#define DEFAULT_Y_ACCELERATION	12.0f	///\$121
#define DEFAULT_Z_ACCELERATION	10.0f	///\$122
#define DEFAULT_A_ACCELERATION	180.0f	///\$123
#define DEFAULT_X_MAX_TRAVEL	360.0f	///\$130
#define DEFAULT_Y_MAX_TRAVEL	233.0f	///\$131
#define DEFAULT_Z_MAX_TRAVEL	90.0f	///\$132
#define DEFAULT_A_MAX_TRAVEL	0.0f	///\$133
#define DEFAULT_X_CURRENT	1200.0f	///\$140
#define DEFAULT_Y_CURRENT	1200.0f	///\$141
#define DEFAULT_Z_CURRENT	700.0f	///\$142
#define DEFAULT_A_CURRENT	700.0f	///\$143
#define DEFAULT_SPINDLE_AT_SPEED_TOLERANCE	0.0f	///\$340
#define DEFAULT_TOOLCHANGE_MODE	0	///\$341
#define DEFAULT_TOOLCHANGE_PROBING_DISTANCE	30	///\$342
#define DEFAULT_TOOLCHANGE_FEED_RATE	25.0f	///\$343
#define DEFAULT_TOOLCHANGE_SEEK_RATE	200.0f	///\$344
#define DEFAULT_TOOLCHANGE_PULLOFF_RATE	200.0f	///\$345
#define DEFAULT_TOOLCHANGE_NO_RESTORE_POSITION	Off	///\$346
#define DEFAULT_TOOLCHANGE_AT_G30	Off	///\$346
#define DEFAULT_TOOLCHANGE_FAST_PROBE_PULLOFF	Off	///\$346
#define DEFAULT_DUAL_AXIS_HOMING_FAIL_AXIS_LENGTH_PERCENT	5.0f	///\$347
#define DEFAULT_DUAL_AXIS_HOMING_FAIL_DISTANCE_MIN	2.5f	///\$348
#define DEFAULT_DUAL_AXIS_HOMING_FAIL_DISTANCE_MAX	25.0f	///\$348
#define DEFAULT_AXIS_ROTATIONAL_MASK	0	///\$376
#define DEFAULT_DISABLE_G92_PERSISTENCE	Off	///\$384
#define DEFAULT_SAFETY_DOOR_SPINDLE_DELAY	4.0f	///\$392
#define DEFAULT_SAFETY_DOOR_COOLANT_DELAY	1.0f	///\$392
#define DEFAULT_SPINDLE_ON_DELAY	0	///\$394
#define DEFAULT_PLANNER_BUFFER_BLOCKS	100	///\$398
#define DEFAULT_AUTOREPORT_INTERVAL	0	///\$481
#define DEFAULT_TIMEZONE_OFFSET	0.0f	///\$482
#define DEFAULT_NO_UNLOCK_AFTER_ESTOP	Off	///\$484
#define DEFAULT_AXIS_ROTARY_WRAP_MASK	0	///\$538
#define DEFAULT_SPINDLE_OFF_DELAY	0	///\$539
#define DEFAULT_HOME_SIGNALS_INVERT_MASK	0	///\$671
#define DEFAULT_COOLANT_ON_DELAY	0	///\$673
#define DEFAULT_HOMING_KEEP_STATUS_ON_RESET	Off	///\$676
#define DEFAULT_KEEP_OFFSETS_ON_RESET	Off	///\$676
#define DEFAULT_STEPPER_ENABLE_DELAY	0	///\$680
#define DEFAULT_MOTOR_WARNING_SIGNALS_ENABLE	0	///\$742
#define DEFAULT_MOTOR_WARNING_SIGNALS_INVERT	0	///\$743
#define DEFAULT_MOTOR_FAULT_SIGNALS_ENABLE	0	///\$744
#define DEFAULT_MOTOR_FAULT_SIGNALS_INVERT	0	///\$745

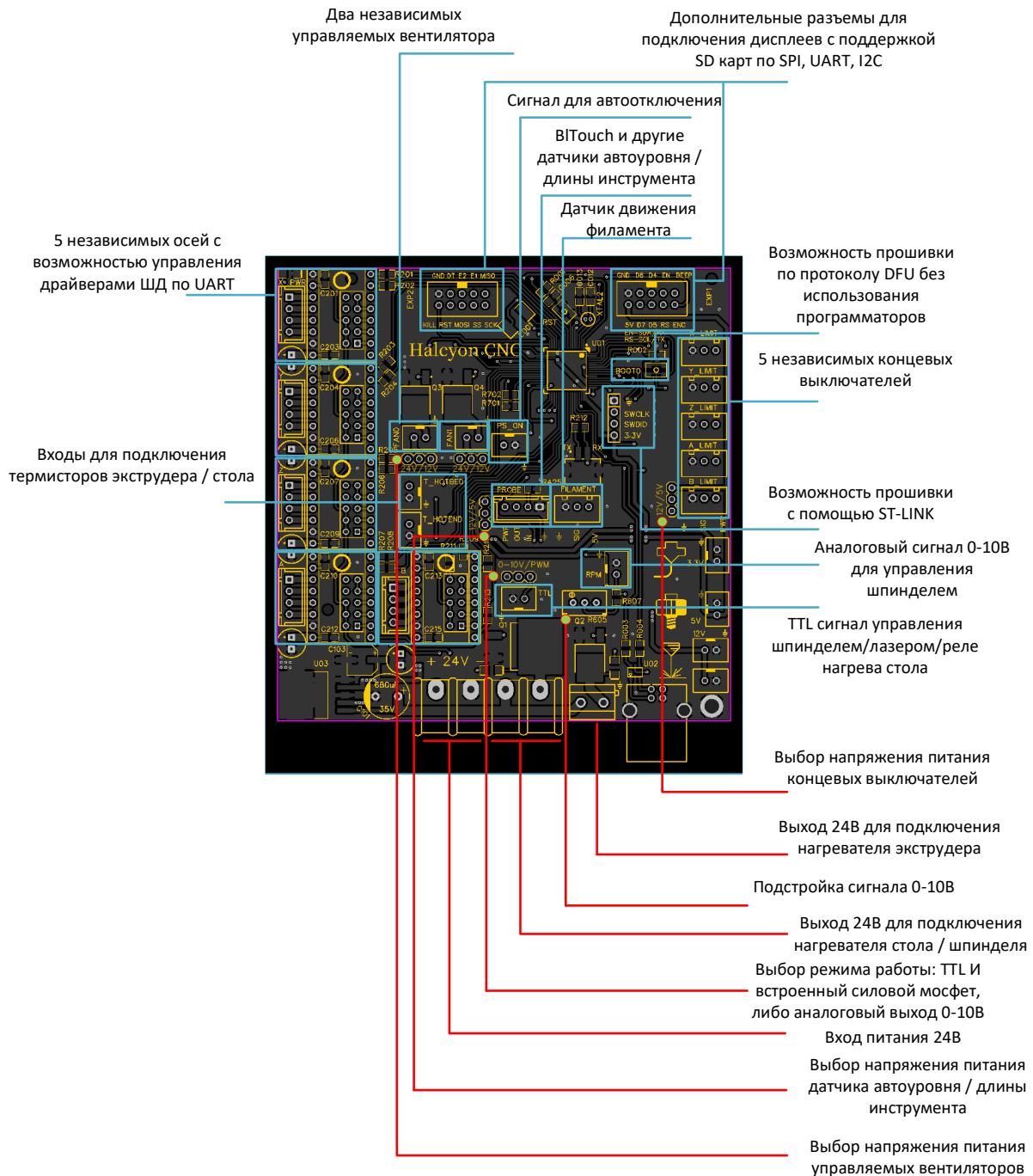
После всей предварительной подготовки следует выполнить компиляцию и загрузку прошивки в память МК. Существует два варианта:

1. Базовый вариант без использования загрузчика. Компилируем прошивку в PlatformIO выбрав окружение [`env:Halcyon_v1_RC`] для STM32F401RCT6 микроконтроллера, либо [`env:Halcyon_v1_RE`] для STM32F401RET6 микроконтроллера. Далее загружаем прошивку во FLASH память МК при помощи ST-Link'a из VSC с установленным PlatformIO или из STM32CubeProgrammer.
2. Расширенный вариант с использованием загрузчика. Такой вариант дает одно преимущество: теперь для обновления прошивки вам достаточно скинуть предварительно скомпилированный файл прошивки на SD карту, подать на плату питание и загрузчик сам обновит программное обеспечение. При использовании загрузчика, первоначально необходимо залить сам загрузчик.

Используйте файл `BOOTLOADER_HALCYON_V1_RC_SPI_SD.bin` / `BOOTLOADER_HALCYON_V1_RE_SPI_SD.bin` (в зависимости от типа микроконтроллера) из репозитория. Предварительно выполните полную очистку FLASH. После загрузите файл bootloader'a при помощи ST-Link и STM32CubeProgrammer. Основная прошивка в данном случае компилируется для окружения [`env:Halcyon_v1_RC_BL`] ([`env:Halcyon_v1_RE_BL`]). Затем вы можете заливать/обновлять прошивку просто записывая файл прошивки с именем **firmware.bin** на SD карту. При подаче питания на плату bootloader проверит SD карту на наличие файла **firmware.bin** и в случае его обнаружения запишет его во FLASH память МК, а после успешной верификации переименует файл в **firmware.cur** и запустит основную прошивку.

Подключение периферийных устройств.

Для подключения периферийных устройств используйте следующую схему:



Выходы HOTBED, TTL, RPM.

Джампер 0-10V/PWM позволяет выбрать режим работы силового выхода. При выборе 0-10V работает только встроенный DAC, и на выходе RPM появляется сигнал с амплитудой от 0 до 10 В. В случае выбора режима PWM, полезный сигнал появляется на разъеме TTL и силовом выходе HOTBED.

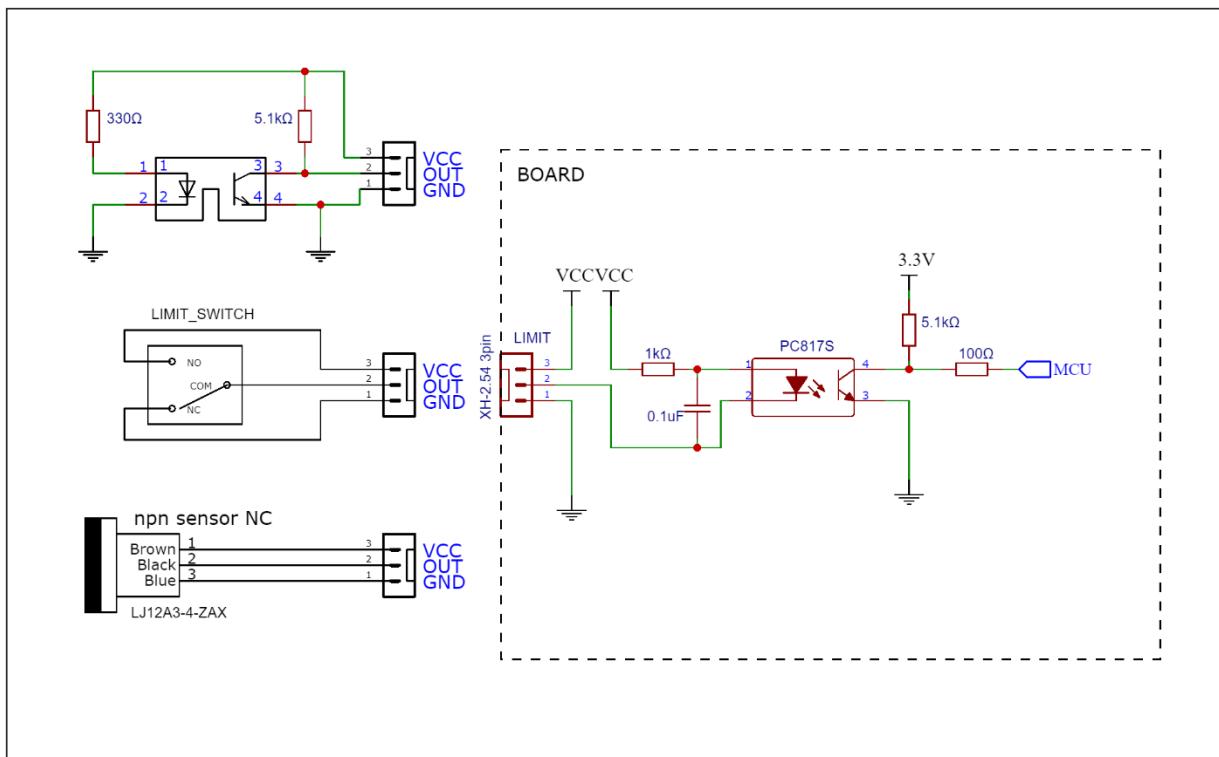
Сигнал	Описание
HOTBED	Силовой ШИМ сигнал с амплитудой, равной напряжению питания платы. Для непосредственного подключения дополнительных устройств с напряжением питания, равным напряжению питания платы, и током до 12А.
TTL	ШИМ сигнал с амплитудой 3.3В и током до 10mA. Для подключения сигнальных цепей реле, в том числе твердотельных.
RPM	Аналоговый сигнал 0-10В. Для подключения внешних драйверов шпинделя.

Точная настройка сигнала RPM:

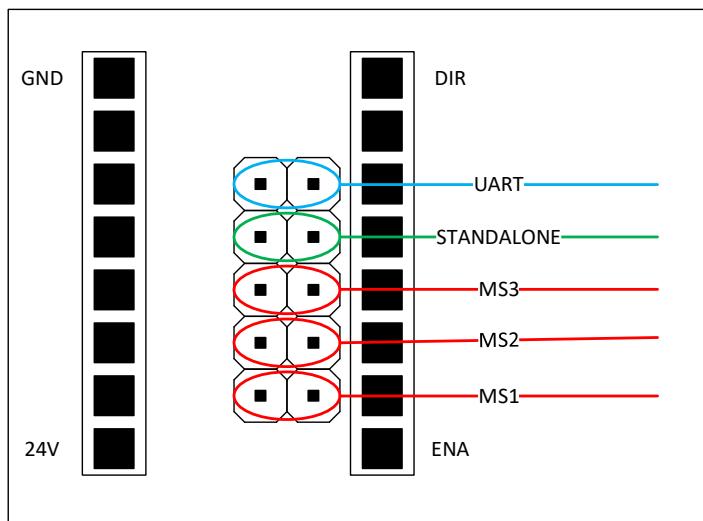
- Подключите драйвер шпинделя и шпиндель. Подключите плату к ПК для возможности внешнего управления.
- Установите потенциометр R605 в крайнее левое положение. В программе управления задайте максимальную скорость вращения шпинделя.
- Вращайте R605 вплоть до момента, когда рост оборотов шпинделя прекратится.

Подключение концевых выключателей.

Плата разработана из расчета надежной работы с нормально замкнутыми на землю (при срабатывании на выходе появляется высокий уровень сигнала) концевыми выключателями любых типов (оптический, индуктивный, механический). При использовании проводников значительной длины в условиях сильных помех добавьте подтягивающий резистор $\sim 10\text{k}\Omega$ со стороны концевого выключателя между выводами OUT и VCC.



Выбор режима работы и микрошага драйверов шаговых двигателей.



Драйвера шаговых двигателей могут работать в режимах STANDALONE и UART (до 4 штук одновременно). Для выбора режима работы, а так же установки микрошага, под каждым драйвером установлено 5 конфигурационных джамперов. Будьте внимательны при установке драйверов, неправильная ориентация может привести к выходу из строя драйвера и платы.

Настройки микрошага:

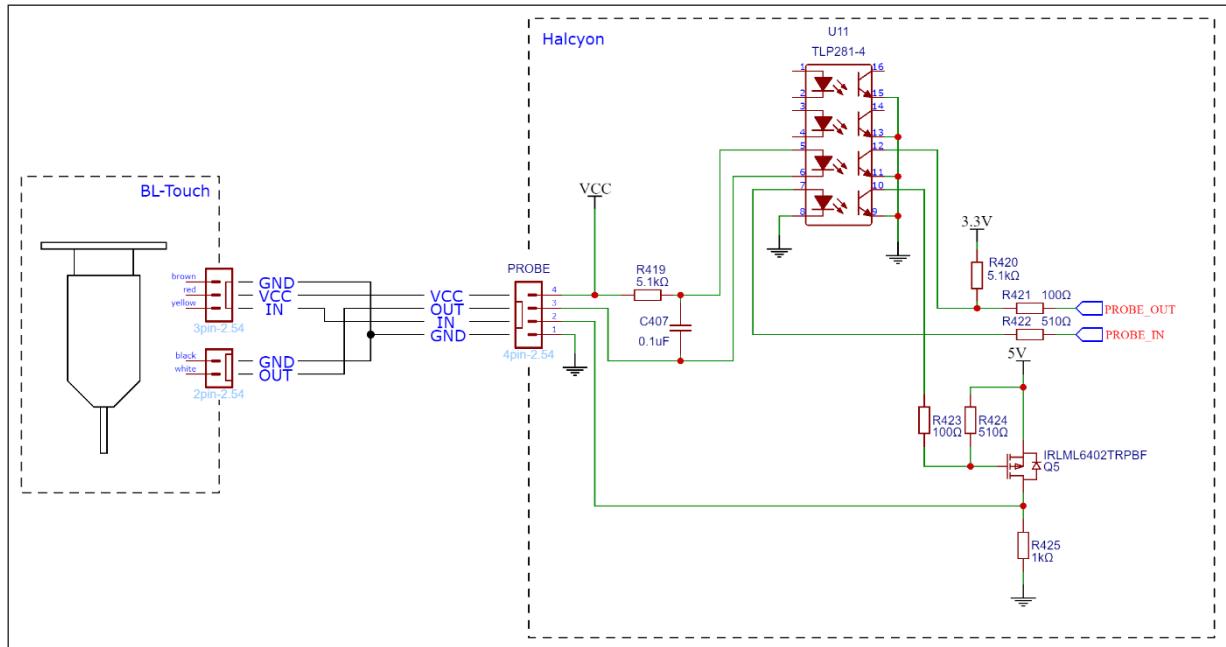
Microstep	A4988			DRV8825			TMC2208		TMC2209		TMC2225		TMC2226	
	MS3	MS2	MS1	MS3	MS2	MS1	MS2	MS1	MS2	MS1	MS2	MS1	MS2	MS1
Full	0	0	0	0	0	0								
1/2	0	0	1	0	0	1	0	1						
1/4	0	1	0	0	1	0	1	0			0	0		
1/8	0	1	1	0	1	1	0	0	0	0	0	1	0	0
1/16	1	1	1	1	0	0	1	1	1	1	1	0	1	1
1/32				1	1	1			0	1	1	1	0	1
1/64									1	0			1	0
1/128														
1/256														

TMC2209 UART адрес:

TMC2209 UART address	Jumpers	
	MS2	MS1
0	0	0
1	0	1
2	1	0
3	1	1

Подключение датчика BL-TOUCH.

Для построения карты высот стола в Marlin вы можете использовать датчик BL-Touch или его аналоги.



Настройки Marlin для построения карты высот стола в файле ..\Marlin\Configuration.h:

```
//#define Z_MIN_PROBEUSES_Z_MIN_ENDSTOP_PIN
//#define USE_PROBE_FOR_Z_HOMING
//#define Z_MIN_PROBE_PIN -1

#define BLTOUCH
#define Z_MIN_PROBE_ENDSTOP_INVERTING false

#define NOZZLE_TO_PROBE_OFFSET { XX.X, YY.Y, ZZ.Z }

#define PROBING_MARGIN 10

#define MULTIPLE_PROBING 3
#define EXTRA_PROBING 1

#define Z_PROBE_OFFSET_RANGE_MIN -5
#define Z_PROBE_OFFSET_RANGE_MAX 5

#define Z_MIN_PROBE_REPEATABILITY_TEST

#define AUTO_BED_LEVELING_UBL

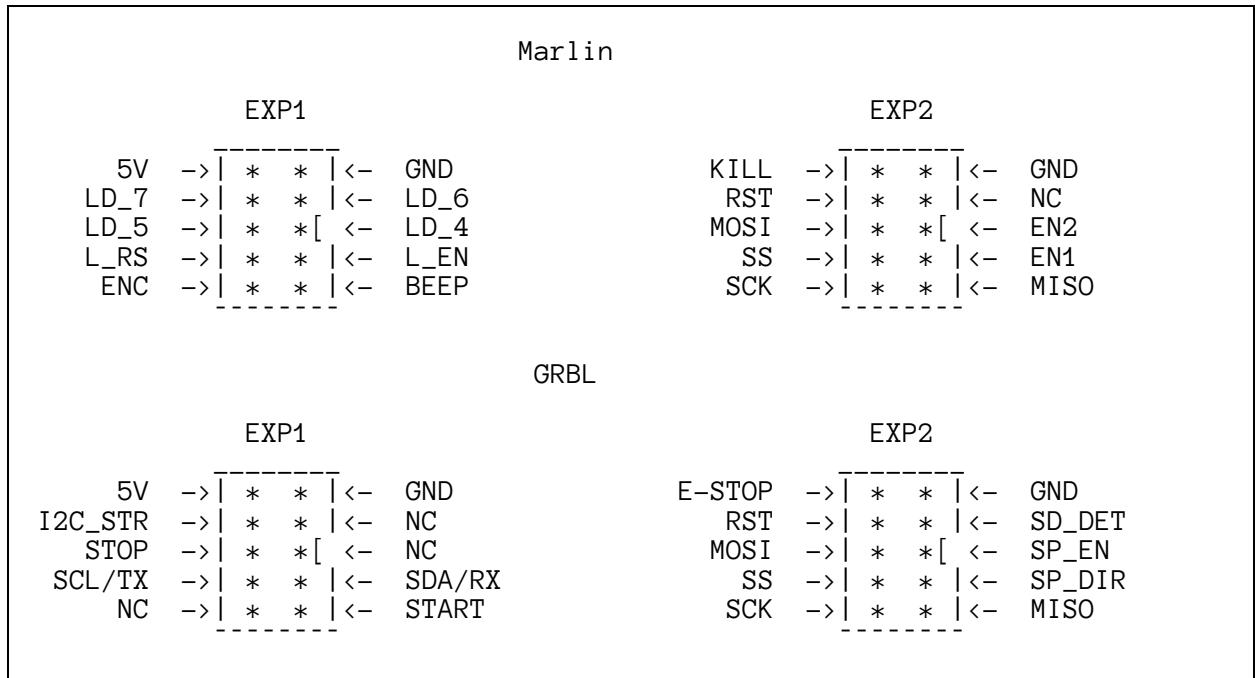
#define Z_SAFE_HOMING

#if ENABLED(Z_SAFE_HOMING)
  #define Z_SAFE_HOMING_X_POINT X_CENTER // (mm) X point for Z homing
  #define Z_SAFE_HOMING_Y_POINT Y_CENTER // (mm) Y point for Z homing
#endif

#elif ENABLED(AUTO_BED_LEVELING_UBL)
  #define MESH_INSET 22
  #define GRID_MAX_POINTS_X 6
  #define GRID_MAX_POINTS_Y 6
```

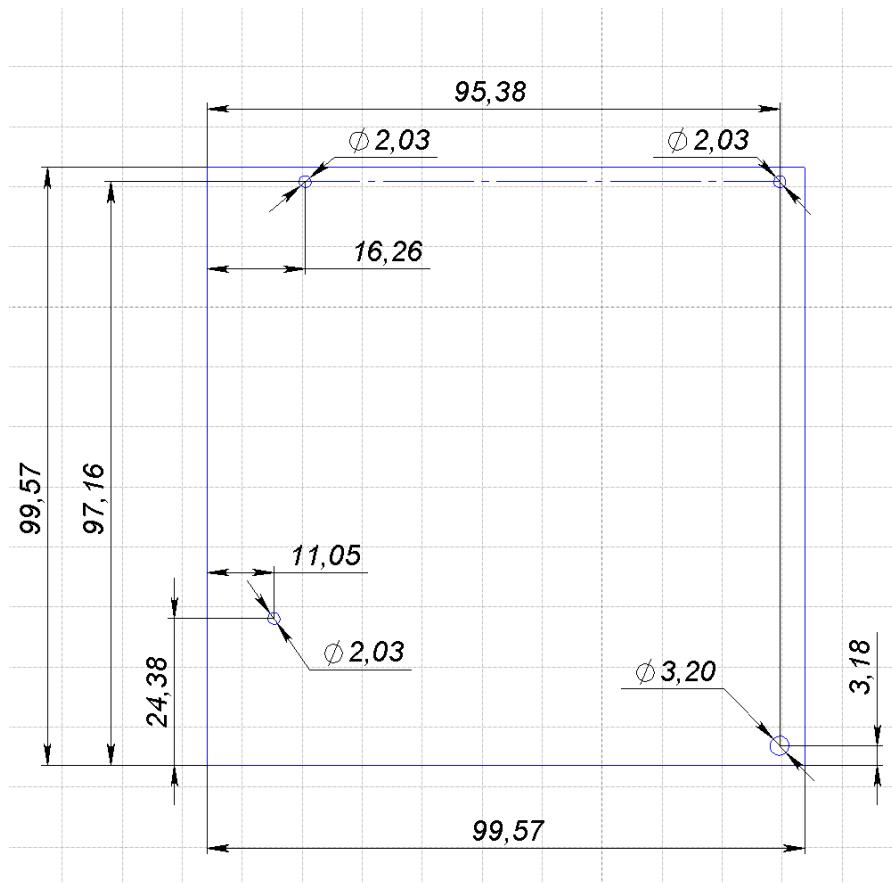
Подключение разъемов дополнительной периферии EXP1 и EXP2.

Вы можете подключить дополнительную периферию в разъемы EXP1 и EXP2 в соответствии со следующей распиновкой:



Дополнительная информация.

Размеры.



Pinout.

```
A0  TM2_CH1 PWM (HOTBED / TTL / 0-10V)
A1  TM2_CH2 PWM (HOTEND)
A2  TM2_CH3 PWM (FAN0 / FLOOD)
A3  TM2_CH4 PWM (FAN1 / MIST)
A4  [EXP 2-07] SD_SS
A5  [EXP 2-09] SCK
A6  [EXP 2-10] MISO
A7  [EXP 2-05] MOSI
A8  Enable motors
A9  TX
A10 RX
A11 USB D-
A12 USB D+
A13 SWDIO
A14 SWCLK
A15 [EXP 1-06] LCD_D4

B0  ADC8 (T_HOTBED)
B1  ADC9 (T_HOTEND)
B2  PS_ON (pulldown)
B3  [EXP 1-03] LCD_D7 / I2C_STROBE
B4  A_MIN
B5  Z_MIN
B6  [EXP 1-07] LCD_RS / SCL / TX
B7  [EXP 1-08] LCD_EN / SDA / RX
B8  X_MIN
B9  Y_MIN
B10 PROBE_IN (servo0)
B11 (N/A)
B12 PROBE_OUT (Z_PROBE)
B13 FILAMENT / DOOR
B14 B_MIN
B15 [EXP 1-08] KILL / E-STOP

C0  X_DIR
C1  X_STEP
C2  Y_DIR
C3  Y_STEP
C4  Z_DIR
C5  Z_STEP
C6  A_DIR
C7  A_STEP
C8  B_DIR
C9  B_STEP
C10 [EXP 1-08] BEEP / CYCLE_START
C11 [EXP 1-08] LCD_D5 / FEED_HOLD
C12 [EXP 1-08] LCD_D6
C13 [EXP 1-09] BTN_ENC
C14 [EXP 2-08] BTN_EN1 / SPINDEL_DIR
C15 [EXP 2-06] BTN_EN2 / SPINDEL_EN

D0  (N/A)
D1  (N/A)
D2  [EXP 2-08] SD_DET
```