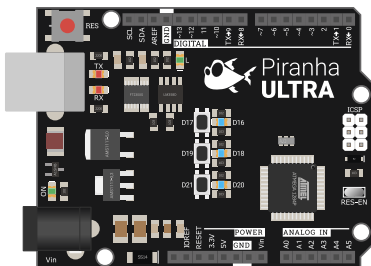


# Piranha Ultra

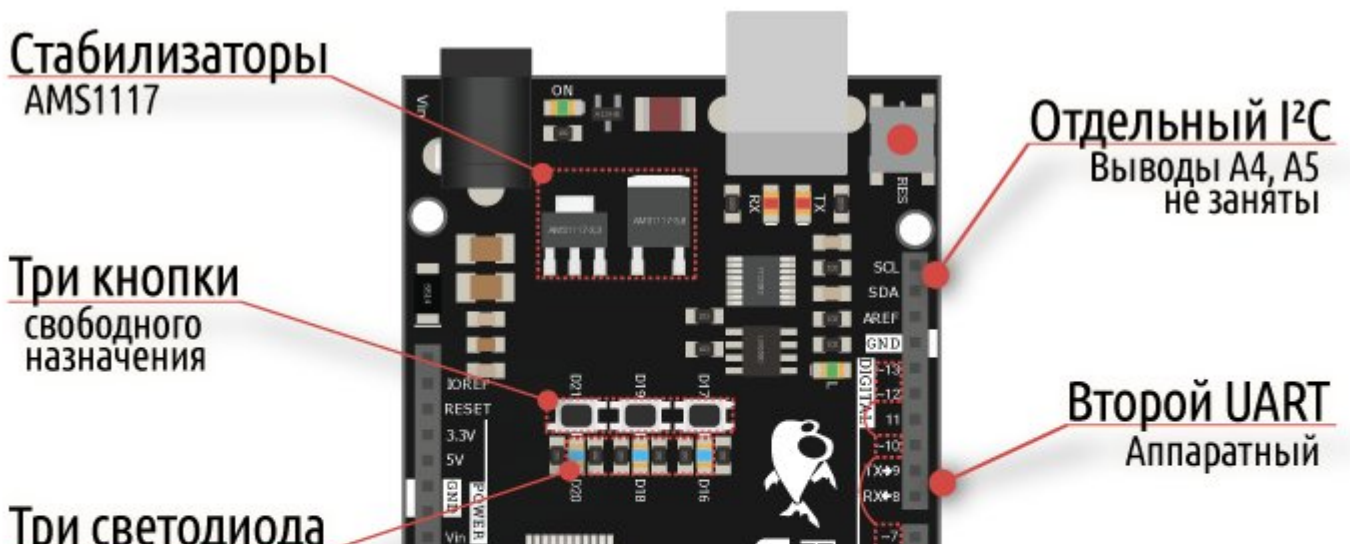


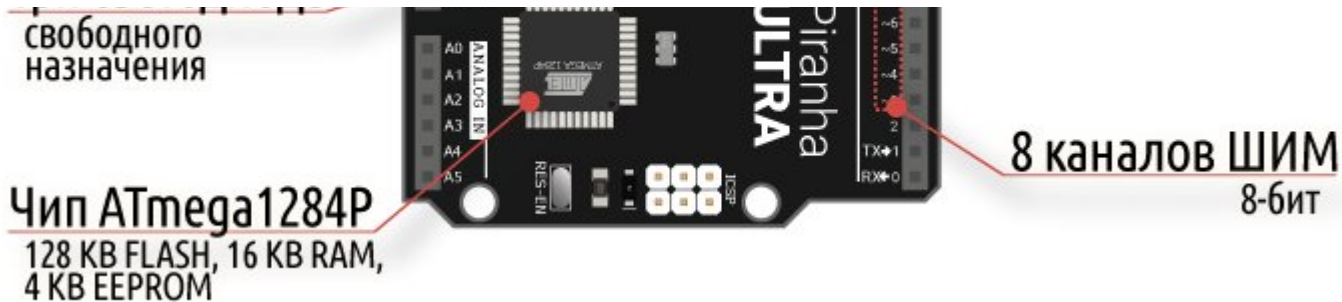
## Общие сведения:

[Piranha ULTRA](#) – разработанная нашей компанией плата практически полностью совместима с проектами на Piranha UNO, при этом память увеличена в 8 раз, имеется два аппаратных UART и шина I2C электрически не связана с выводами A4, A5. Это означает что можно использовать все 6 АЦП и шину I2C одновременно. Так же выводы I2C можно использовать как дополнительные цифровые выводы D14, D15. На плате используются такие же стабилизаторы напряжения для шин питания на 5В и 3.3В, что и на Piranha UNO R3, что положительно сказывается на стабильности работы большинства модулей.

## Спецификация:

- Микроконтроллер: ATmega1284P.
- Ядро: AVR.
- Тактовая частота 16 МГц.
- Разрядность: 8 бит.
- Входное напряжение питания:
  - через разъем питания: 7-12 В.
  - через порт USB: 5 В.
  - источник питания выбирается автоматически (приоритет у разъема питания).
- Уровень логической «1» на выводах (логика чипа): 5 В.
- Максимальный выходной ток на одном выводе I/O: до 20 мА.
- Максимальный суммарный выходной ток выводов I/O: до 150 мА.
- Максимальный ток на выходе 3,3V: 500 мА.
- Максимальный ток на выходе 5V: 800 мА.
- Количество цифровых выводов I/O: 14 выводов (из них 8 выводов поддерживают ШИМ 8 бит).
- Количество аналоговых входов: 6 выводов (АЦП 10 бит).  
(аналоговые входы A0-A5 могут работать как цифровые I/O: D14-D19).
- Объем памяти программ (FLASH): 128 Кбайт (из них 0.5 Кбайт используются под загрузчик).
- Объем оперативной памяти (SRAM): 16 Кбайт.
- Объем ПЗУ (EEPROM): 4 Кбайт.
- Встроенные интерфейсы (аппаратные шины):
  - 1x I2C;
  - 1x SPI;
  - 2x UART;
- 3 кнопки на плате, выводы D17, D19, D21.
- 3 светодиода на плате, выводы D16, D18, D20.
- Преобразователь интерфейсов USB-UART: FTDI.
- Стабилизаторы питания: AMS1117.
- Длина: 68,6 мм
- Ширина: 53,4 мм
- Вес: 25 г.





## Подключение:

Для подключения датчиков и модулей к Piranha UNO R3 удобно использовать [Trema Shield](#) или [Trema Set Shield](#).

Для подключения к компьютеру понадобится [кабель USB](#).

## Питание:

Плата Piranha Uno R3 может быть запитана от USB порта компьютера, или от внешнего источника питания: [AC/DC адаптера на 9В](#), [AC/DC адаптера на 12В](#), штекер которого (Ø 2,1 мм, центральный вывод - положительный) необходимо подключить к силовому разъему питания. При создании мобильных устройств рекомендуем использовать [Battery Shield](#) в качестве источника автономного питания. Если внешним источником питания служит [аккумуляторная батарея](#) от 6,5 до 12В, то её нужно подключить к выводам Vin и GND. Тип источника питания выбирается автоматически, для этого на плате Piranha ULTRA R3 установлен компаратор.

## Сравнение с UNO:

	Piranha UNO	Piranha ULTRA
Совместимость с шилдами UNO	Да	Да
Стабилизаторы питания	AMS1117	AMS1117
Количество каналов ШИМ	6	8
Количество аппаратных шин UART	1	2
Память FLASH	32 Кб	128 Кб
ОЗУ	2 Кб	16 Кб
ПЗУ	1 Кб	4 Кб
Конфигурация шины I2C	Связана с выводами A4, A5	Не связана с выводами A4, A5; Можно использовать как дополнительные цифровые выводы

	Piranha UNO	Piranha ULTRA
		D14, D15
Дополнительно	-	Три светодиода и три кнопки свободного назначения на плате

## Подробнее о плате:

Разработанная нами плата Piranha ULTRA R3 является Arduino совместимой платой, она создана на базе микроконтроллера ATmega1284P. Это значит что плата Piranha ULTRA R3 может использоваться практически в любом проекте, созданном для Arduino UNO.

Если Вы хотите использовать много модулей в одном проекте, но вам нужен форм-фактор Arduino UNO, то приобретение платы Piranha ULTRA R3 будет оптимальным вариантом среди всей линейки плат Arduino, так как у платы 16 Кбайт динамической и 128 Кбайт FLASH памяти, что позволяет использовать большее количество библиотек одновременно.

На плате Piranha ULTRA R3 используется FTDI преобразователь USB-UART, значит Вам не потребуется устанавливать драйвера при подключении платы к USB порту компьютера с ОС Windows 10, плата определится как FT230x, а в Arduino IDE как COMxx (Например COM10).

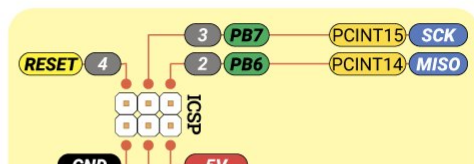
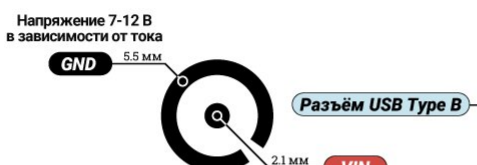
Светодиоды платы Piranha UNO R3 размещены так, что Вы сразу поймёте, за что они отвечают:

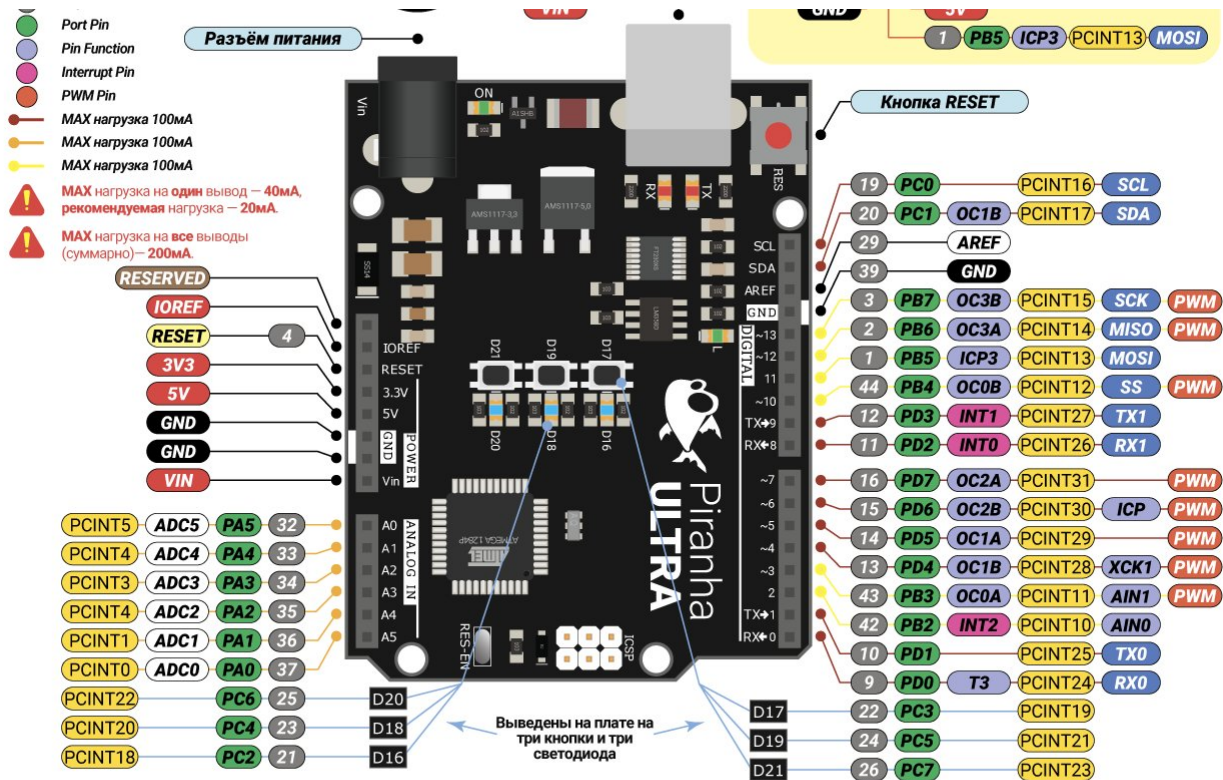
- **Светодиод ON**, сигнализирующий о наличии питания, расположен рядом с разъёмом питания;
- **Светодиоды RX и TX** преобразователя USB-UART расположены рядом с USB разъемом и сигнализируют о передаче данных к/от компьютера;
- **Светодиод L**, подключённый к выводу D13, расположен рядом с этим выводом и информирует о наличии уровня логической «1» на нём;
- **3 светодиода** свободного назначения на выводах D16, D18, D20.

Сборка плат Piranha ULTRA R3 осуществляется на нашем производстве, где каждая плата проходит контроль качества.

На плате Piranha ULTRA R3 имеется колодка питания, колодки аналоговых и цифровых выводов для подключения внешних модулей, использующих 5В логику. К этим выводам вы можете подключать сенсоры, датчики, дисплей, кнопки, индикаторы, драйверы, реле, shield и т.д. После чего из компьютера в микроконтроллер через порт USB загружается программа (скетч) которая реализует работу всех подключённых модулей по заложенному в неё алгоритму.

## Карта выводов Piranha ULTRA





## Пример использования встроенных кнопок:

Данный пример включает встроенные светодиоды при нажатии на строенные кнопки.

```

void setup() {
// Конфигурируем выводы кнопок:
pinMode(KEY_BUILTIN_1, INPUT ); // Указываем выводу 1 кнопки работать как в
pinMode(KEY_BUILTIN_2, INPUT ); // Указываем выводу 2 кнопки работать как в
pinMode(KEY_BUILTIN_3, INPUT ); // Указываем выводу 3 кнопки работать как в
// Конфигурируем выводы светодиодов:
pinMode(LED_BUILTIN_1, OUTPUT); // Указываем выводу 1 светодиода работать к
pinMode(LED_BUILTIN_2, OUTPUT); // Указываем выводу 2 светодиода работать к
pinMode(LED_BUILTIN_3, OUTPUT); // Указываем выводу 3 светодиода работать к
}

void loop() {
// Читаем состояния кнопок:
bool a = digitalRead( KEY_BUILTIN_1 ); // Читаем состояние 1-й кнопки в переменную
bool b = digitalRead( KEY_BUILTIN_2 ); // Читаем состояние 2-й кнопки в переменную
bool c = digitalRead( KEY_BUILTIN_3 ); // Читаем состояние 3-й кнопки в переменную
// Управляем светодиодами:
digitalWrite(LED_BUILTIN_1, a); // Включаем 1-й светодиод если установлен ф
digitalWrite(LED_BUILTIN_2, b); // Включаем 2-й светодиод если установлен ф
digitalWrite(LED_BUILTIN_3, c); // Включаем 3-й светодиод если установлен ф
}

```

Следующий пример демонстрирует работу со встроенными кнопками и светодиодами используя номера их выводов (D16-D21) вместо констант KEY\_BUILTIN и LED\_BUILTIN.

```
#define BUTTON1 17 // Определяем вывод 1-й кнопки
#define BUTTON2 19 // Определяем вывод 2-й кнопки
#define BUTTON3 21 // Определяем вывод 3-й кнопки
//
#define LED1 16 // Определяем вывод 1-го свето
#define LED2 18 // Определяем вывод 2-го свето
#define LED3 20 // Определяем вывод 3-го свето
//
bool LED1_STATE = LOW; // Обозначаем состояние 1-го с
bool LED2_STATE = LOW; // Обозначаем состояние 2-го с
bool LED3_STATE = LOW; // Обозначаем состояние 3-го с
//
// Функция переключения состояния светодиодов: //
void button_press(int _button, bool& _led_state){ // Принимаем номер кнопки и со
    if (digitalRead(_button)){ // Считываем состояние кнопки,
        delay(100); // Ждём 100 миллисекунд, для у
        if (digitalRead(_button)){ // Если кнопка до сих пор зажа
            _led_state = !_led_state; // меняем состояние светодиода
        } //
    } //
} //
//
void setup() { //
    pinMode(LED1, OUTPUT); pinMode(BUTTON1, INPUT); // Устанавливаем режим работы
    pinMode(LED2, OUTPUT); pinMode(BUTTON2, INPUT); // Устанавливаем режим работы
    pinMode(LED3, OUTPUT); pinMode(BUTTON3, INPUT); // Устанавливаем режим работы
} //
//
void loop() { //
    button_press(BUTTON1, LED1_STATE); // Вызываем функцию button_pre
    button_press(BUTTON2, LED2_STATE); // Вызываем функцию button_pre
    button_press(BUTTON3, LED3_STATE); // Вызываем функцию button_pre
//
    digitalWrite(LED1, LED1_STATE); // Устанавливаем логический ур
    digitalWrite(LED2, LED2_STATE); // Устанавливаем логический ур
    digitalWrite(LED3, LED3_STATE); // Устанавливаем логический ур
//
}
```

## Пример использования второго аппаратного UART:

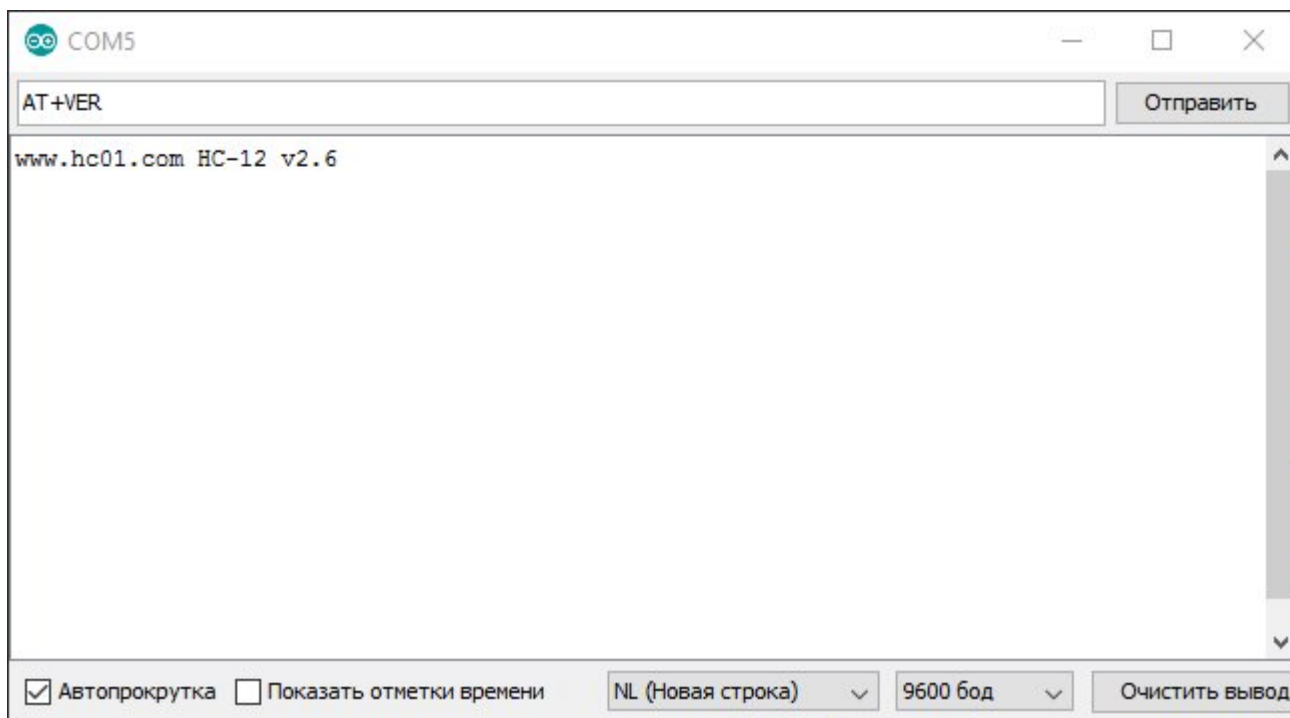
Piranha ULTRA имеет два аппаратных UART. Первый, как и у Piranha UNO, находится на выводах 0, 1 и используется для общения с серийным монитором Arduino IDE, а второй находится на выводах 8, 9 и может использоваться независимо от первого. Например, можно принимать информацию в серийный монитор по UART0 при это общаясь с серийным устройством по UART1. В



данном примере это радио-модуль HC-12.

```
#define SetPin 7 // Определяем вывод активации AT команд
//
//
void setup() { //
  Serial.begin(9600); // Включаем серийный порт 0 на скорости 9600 бод
  Serial1.begin(9600); // Включаем серийный порт 1 на скорости 9600 бод
  pinMode(SetPin, OUTPUT); // Устанавливаем вывод в режим выход
  digitalWrite(SetPin, LOW); // Устанавливаем логический ноль на выводе
} //
//
void loop() { //
  if(Serial1.available()){ // Если в буфере порта 1 есть данные
    Serial.write(Serial1.read()); // отправляем их в порт 0
  } //
  if(Serial.available()){ // Если в буфере порта 0 есть данные
    Serial1.write(Serial.read()); // отправляем их в порт 1
  } //
} //
```

Теперь мы можем посылать AT команды модулю через аппаратный UART:



## Поддержка плат Piranha в Arduino IDE

После подключения платы к компьютеру в Arduino IDE вы увидите новый порт, который не будет никак подписан. Для того, чтобы добавить поддержку плат семейства Piranha в Arduino IDE, достаточно выполнить несколько простых шагов, описанных в [пошаговой инструкции](#). Если Вы уже настраивали поддержку плат семейства Piranha, не забудьте обновить её до последней версии в

менеджере плат Arduino IDE (Меню: "Инструменты -> Плата: -> Менеджер плат...").

## Программное обеспечение:

Скачать (загрузить) программу Arduino IDE для создания, редактирования и загрузки скетчей в Piranha ULTRA R3, Вы можете с [официального сайта](#).

Для работы с платой Piranha ULTRA R3 в программе Arduino IDE необходимо указать, что используется плата Piranha ULTRA.

