

## **Инструкция для участников хакатона по отправке пакетов данных в UART и использованию COMReader**

Одна из важных задач для успешного прохождения хакатона – правильная передача обработанных данных, так как, если вы написали идеальный алгоритм, но не смогли корректно передать результаты, судья не сможет по достоинству оценить вашу работу. Поэтому в данной инструкции, мы опишем требования по передаче данных.

### **Интерфейс передачи и его настройка:**

- 1) Для передачи данных необходимо использовать интерфейс UART настроенный на baud rate = 115200 Bits/s, количество бит на символ – 8 bit, количество стоповых битов – 1, контроль четности отсутствует.
- 2) Настраивать UART необходимо по прерыванию на прием и на передачу.

### **Условия на начало и конец передачи данных:**

- 1) Начинать отправку данных необходимо после получения символа: 'r' – uint8\_t (ASCII – 114).
- 2) Остановить передачу необходимо после получения символа: 's' – uint8\_t (ASCII – 115).

### **Структура отправляемого пакета данных:**

Наименование параметра	Тип данных	Размер, байт	Размерность, значение
Заголовок пакета	UINT32	4	0x0200A6BD
Магнитная интенсивность X	FLOAT32	4	Микро Тесла
Магнитная интенсивность Y	FLOT32	4	Микро Тесла
Магнитная интенсивность Z	FLOAT32	4	Микро Тесла
Ускорение X	FLOAT32	4	м/с/с
Ускорение Y	FLOAT32	4	м/с/с

Ускорение Z	FLOAT32	4	м/с/с
Угловая скорость X	FLOAT32	4	°/ сек
Угловая скорость Y	FLOAT32	4	°/ сек
Угловая скорость Z	FLOAT32	4	°/ сек

### Проверка приема-передачи:

COMReader – это Python script, который позволяет обрабатывать пакеты данных, поступающих через COM порт ПК. Script необходим Вам для проверки и отладки корректной отправки и получения данных. При проверке судьей Вашей работы будет использоваться именно этот script.

### Требования для использования COMReader:

- 1) Для использования COMReader Вам желательно иметь на ПК настроенную IDE для работы с языком программирования Python (PyCharm, VS Code или др.).

### Начало работы с COMReader

- 1) Подключите Ваше устройство к USB порту вашего ПК.
- 2) Определите с помощью диспетчера устройств COM порт, к которому Ваше устройство привязано.
- 3) Откройте с помощью IDE файл COMReader\_for\_participants.py
- 4) Для запуска файла COMReader\_for\_participants.py Вам необходимо будет установить библиотеки math, struct, keyboard, serial.
- 5) В поле port ввести номер COM порта, который узнали из диспетчера устройств.

```
USER = serial.Serial(
    port="COM13", baudrate=115200, bytesize=8,
    timeout=100, stopbits=serial.STOPBITS_ONE
```

- 6) После настройки можно запускать script.

### Запуск и работа:

- 1) Для начала получения данных с Вашего устройства нажмите на клавиатуре клавишу 'r'.
- 2) После этого в консоль выведется лист с данными, которые пришли с Вашего устройства (при условии, что все настроено верно). Также создастся файл USER\_data.txt, в котором будут те же самые данные, что и в консоли.

При возникновении вопросов писать в [Telegram](#).

P.S. Пример корректно работающей реализации приема-передачи на STM32:

```
uint8_t flag = 0;
uint8_t command = 0;

void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    if (huart == &huart1)
    {
        if (command == 'r') flag = 1;
        if (command == 's') flag = 0;
    }
    HAL_UART_Receive_IT(&huart1, &command, 1);
}

int main()
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_USART1_UART_Init();

    float data[9] = {0.2348, 1.5478, 88.9567, 100.4589, 3.0234,
7.0005, 9.8907, 23.2951, 4.9084 };
    uint32_t a = 3181772803;
    uint8_t sw[4] = { 0x02, 0x00, 0xA6, 0xBD };
    uint8_t data8[36] = {0, };
    HAL_UART_Receive_IT(&huart1, &command, 1);

    memcpy(data8, (uint8_t*)data, sizeof(data));
```

```
while (1)
{
    if (flag)
    {
        HAL_UART_Transmit_IT(&huart1, sw, 4);
        HAL_Delay(10);
        HAL_UART_Transmit_IT(&huart1, (uint8_t*)data,
sizeof(data));
        HAL_Delay(10);
    }
}
}
```