Учимся общаться с компьютером.

Мы научились взаимодействовать с контроллером «РУДИРОН» посредством кнопок и светодиодов.

В составе контроллера есть универсальный асинхронный приёмопередатчик (Univsersal Asynchronos Reciever-Transmitter UART) - это физическое устройство приёма и передачи данных по двум проводам. Оно позволяет двум устройствам обмениваться данными на различных скоростях.

UART позволяет используя минимум проводов для связи двух устройств. Но только двух устройств. Очень часто контроллер собирает данные с различных датчиков и по UART передает данные в компьютер для последующей обработке.

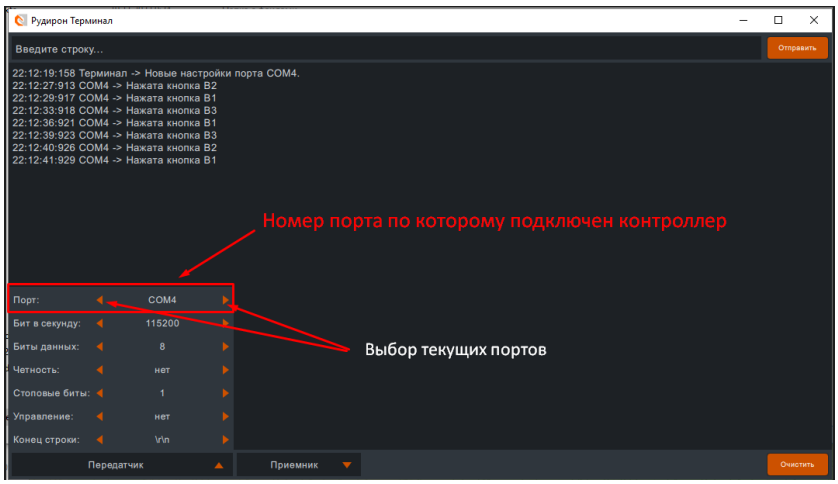
Задача – программа реагирует на нажатие кнопок B1, B2, B3 и выводит в UART сообщение о том какая кнопка нажата. Сообщение принимается терминалом на компьютере и печатается пользователю на экране.

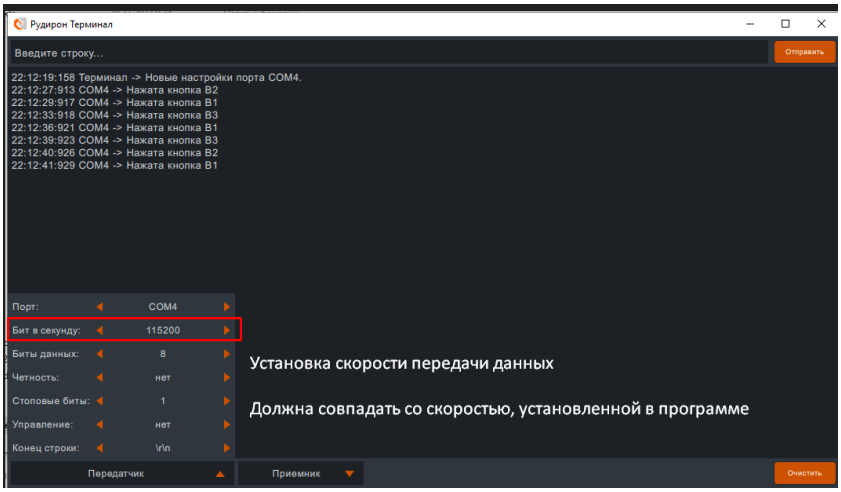
```
void setup()
{
  //конфигурация встроенных кнопок
  pinMode(BUTTON_BUILTIN_1, INPUT_PULLDOWN);
  pinMode(BUTTON_BUILTIN_2, INPUT_PULLDOWN);
  pinMode(BUTTON_BUILTIN_3, INPUT_PULLDOWN);

  //конфигурация встроенных светодиодов
  pinMode(LED_BUILTIN_1, OUTPUT);
  pinMode(LED_BUILTIN_2, OUTPUT);

  //включение встроенных светодиодов
  digitalWrite(LED_BUILTIN_1, true);
  digitalWrite(LED_BUILTIN_2, true);

  //конфигурация последовательного порта
  Serial.begin(115200);
  //отправка приветствия через последовательный порт
  Serial.println("Рудирон Бутерброд!");
}
```

```
void loop()
{
   //чтение встроенных кнопок, true = есть нажатие
   bool pressed1 = digitalRead(BUTTON_BUILTIN_1);
   bool pressed2 = digitalRead(BUTTON_BUILTIN_2);
   bool pressed3 = digitalRead(BUTTON_BUILTIN_3);

        if (pressed1) { Serial.println("Нажата кнопка B1");}
        if (pressed2) { Serial.println("Нажата кнопка B2");}
        if (pressed3) { Serial.println("Нажата кнопка B3");}



         //пауза программы на 1 секунду
        delay(1000);
}
```

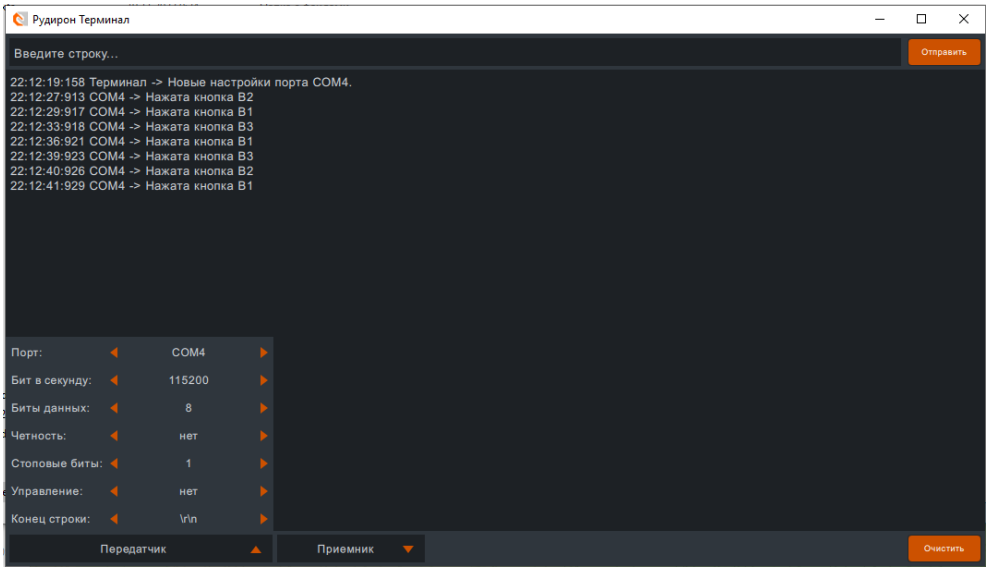
Для общения компьютера с контроллером запустим программу терминала D:\dibot\projects\tools\Rudiron Terminal.exe. Терминал находится в папке с дистрибутивом внутри директории tools.

Чтобы компьютер «увидел» наш контроллер необходимо установить скорость обмена и COM порт по которому идет подключение.

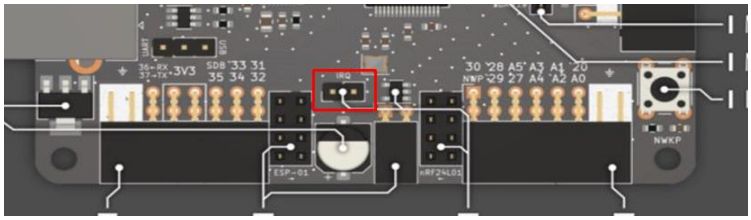




После установки необходимых параметров можно нажимать на кнопки B1, B2, B3. В главном окне будут появляться сообщения, когда вы нажимаете на одну из кнопок.



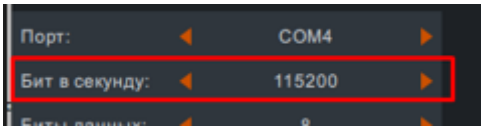
Может быть ситуация, что будет выводиться сообщение о постоянном нажатии кнопки B2. Это связано с перемычкой IRQ. Снимите ее и все будет работать корректно.



Рассмотрим, что у нас нового в программе:

1. Установка скорости обмена Serial.begin(115200);

Данную скорость выставляем в настройках терминала на компьютере.



2. Даем контроллеру команду на передачу данных в виде текстовой строки Serial.println("Нажата кнопка B1");

Очень простая возможность передавать данные в компьютер. Часто используют данную передачу данных из разных частей программы, чтобы понять в какие значения хранятся в тех или иных переменных. Такой прием позволяет отладить программу если не понятно почему она работает не корректно.

Как пример передадим в компьютер значение переменной angle.

1. Создаем переменную:

```
int angle = 20;
```

Переменной сразу присвоили значение 20.

2. Передаем сначала информационную строку о переменной в виде строки:

   Serial.print("Переменная angle =");

   Обратите внимание, что используем print вместо printl. Printl печатает сообщение и переводит указатель на строку ниже. Поэтому следующее сообщение печатается под первым. Print не переводит указатель поэтому следующее сообщение будет распечатано в этой же строке.

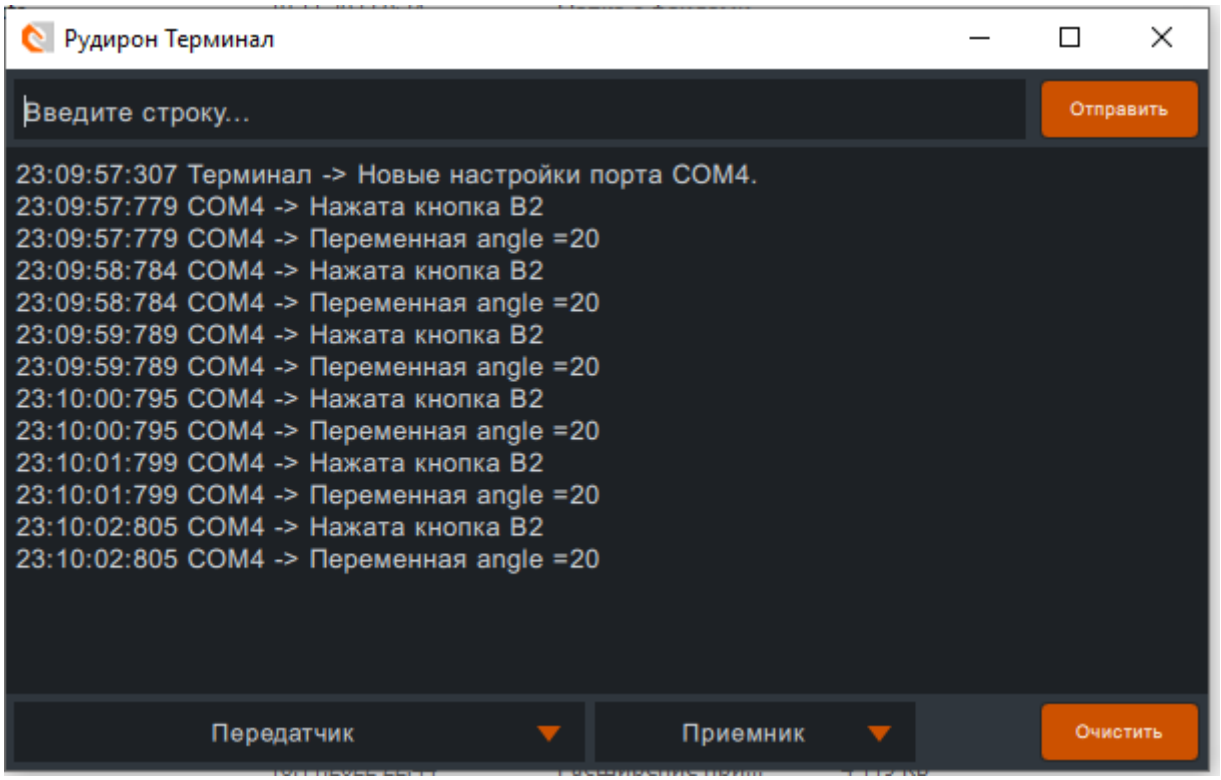
3. Передаем в UART значение самой переменной Serial.println(angle, DEC). Помимо указания самой переменной необходимо указать константу форматирования – DEC десятичное число.

   Варианты значений констант для форматирования:

   - DEC – обычное число в десятичной системе исчисления
   - BIN – преобразует в двоичный код и выведет строку, содержащую только символы 0 и 1

- OCT – преобразует в восьмеричную систему исчисления
- HEX – преобразует в шестнадцатеричную систему
- Цифра от 0 до 9 – используется, если первый аргумент – вещественное число с плавающей запятой. Форма указывает количество знаков после запятой, которые останутся при выводе. Само число при этом будет округлено

В итоге в терминале мы увидим вот такой результат. Обратите внимание на кнопку B2. Мы про нее писали выше.



Терминал и сам UART позволяет нам передавать в контроллер информацию. Напишем программу, которая принимает от компьютера 1 или 0. Если приняли 1, то включаем светодиод L1. Если принимаем 0, то выключаем L1.

```
byte inputByte;

void setup()
{

    pinMode(BUTTON_BUILTIN_1, INPUT_PULLDOWN);
```

```
	pinMode(LED_BUILTIN_1, OUTPUT);

	digitalWrite(LED_BUILTIN_1, true);

	//конфигурация последовательного порта
	Serial.begin(115200);

}

void loop()
{
	if (Serial.available() > 0)
	{
		inputByte = Serial.read();

		if (inputByte == '1')
		{
			digitalWrite(LED_BUILTIN_1, HIGH);
		}
		else if (inputByte == '0')
		{
			digitalWrite(LED_BUILTIN_1, LOW);
		}

		Serial.print("Я принял следующие символы: ");
		Serial.println(inputByte, DEC);
	}
	delay(10);
}
```

Новые строчки в программе:

1. Serial.available() > 0 – проверка был ли принят хоть один символ от компьютера во внутреннюю память UART. Если принят, то можно его считать в переменную.

2. inputByte = Serial.read() – заносим в переменную считанный символ из внутренней памяти UART.

3. Если принятый символ 1, то подаем на светодиод L1 3.3 вольта (светится):

```
		if (inputByte == '1')
				{
```

```
                digitalWrite(LED_BUILTIN_1, HIGH);
                }
```

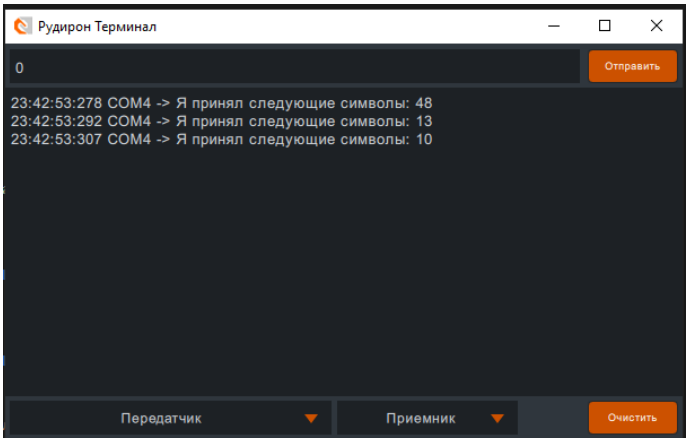
4. Если принятый символ 0, то подаем ноль вольт на светодиод L1 (не светится):

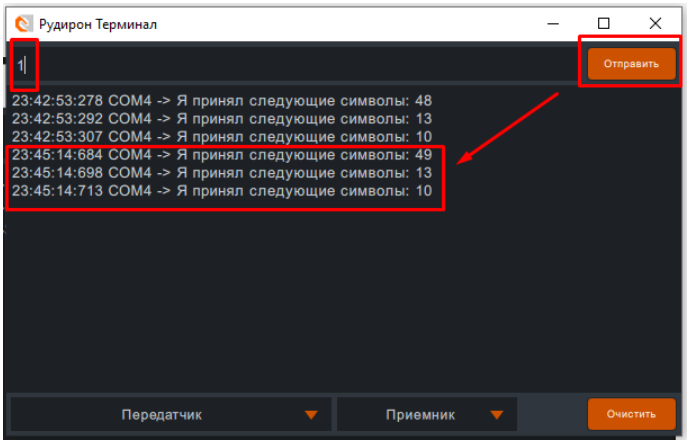
```
else if (inputByte == '0')
        {
            digitalWrite(LED_BUILTIN_1, LOW);
        }
```

5. Выводим обратно в компьютер строки, чтобы увидеть какие символы были приняты нашим контроллером от компьютера:

```
Serial.print("Я принял следующие символы: ");
Serial.println(inputByte, DEC);
```

Посмотрим какой результат буде в терминале и на контроллере. Для передачи символов в контроллер в верхнем поле ввода необходимо ввести символ с клавиатуры и нажать на клавишу отправить.





Программа работает корректно. При передаче символа 1 светится светодиод L1 при передаче символа 0 он гаснет.

Однако можно заметить, что контроллер принимает не один символ, а сразу 3. Это связано с системой передачи строковых данных через UART. Первый символ у нас не 1 или 0, а 49 и 48. Это связано с тем, что передаются

коды символов, а не десятичные числа 0 или 1. За каждой клавишей на клавиатуре (вернее символом) зарезервирован свой код.

Escape 27		F1 112	F2 113	F3 114		F4 115	F5 116	F6 117	F7 118	F8 119	F9 120	F10 121	F11 122	F12 123	Print Screen	Scroll Lock 145	Pause 19			
`ё 192	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	0 48	- 189	=+ 187	Back Space 8	Insert 45	Home 36	Page Up 33	Num Lock 144	/ доп. 111	* доп. 106	+ доп. 107
Tab 9	Q 81	W 87	E 69	R 82	T 84	Y 89	U 85	I 73	O 79	P 80	[219	] 221		Delete 46	End 35	Page Down 34	7 доп. 103	8 доп. 104	9 доп. 105	
Caps Lock 20	A 65	S 83	D 68	F 70	G 71	H 72	J 74	K 75	L 76	;ж 186	'э 222	Enter 13					4 доп. 100	5 доп. 101	6 доп. 102	
Shift 16	Z 90	X 88	C 67	V 86	B 66	N 78	M 77	,< 188	.> 190	/ 191	Shift 16	| 220			Up 38		1 доп. 97	2 доп. 98	3 доп. 99	Enter доп. 13
Ctrl 17	win	Alt 18	Space Bar 32						Alt 18	win	list	Ctrl 17		Left 37	Down 40	Right 39	Ins/0 45/96		Del/. 46/110	

Раньше текст печатали на печатных машинках. И когда текст

Когда последняя буква в строке была напечатана необходимо было перейти на строку ниже и передвинуть бумагу так, чтобы следующий символ оказался с левого края бумаги. Это две операции символ 13 - это carriage return (возврат каретки return/ввод) и символ 10 - это line feed (новая строка). Вот как исторически сложилась система передачи информации.