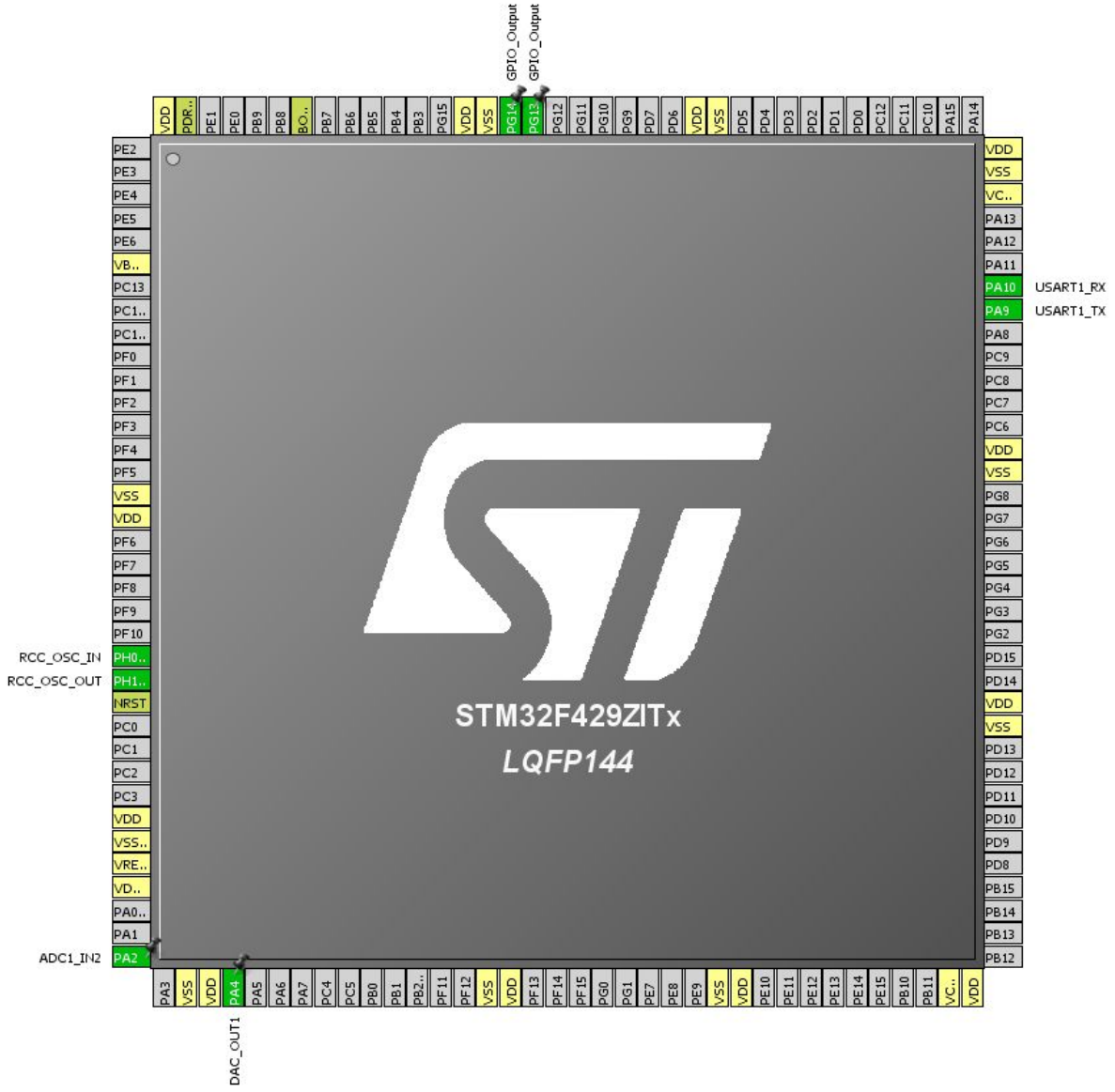
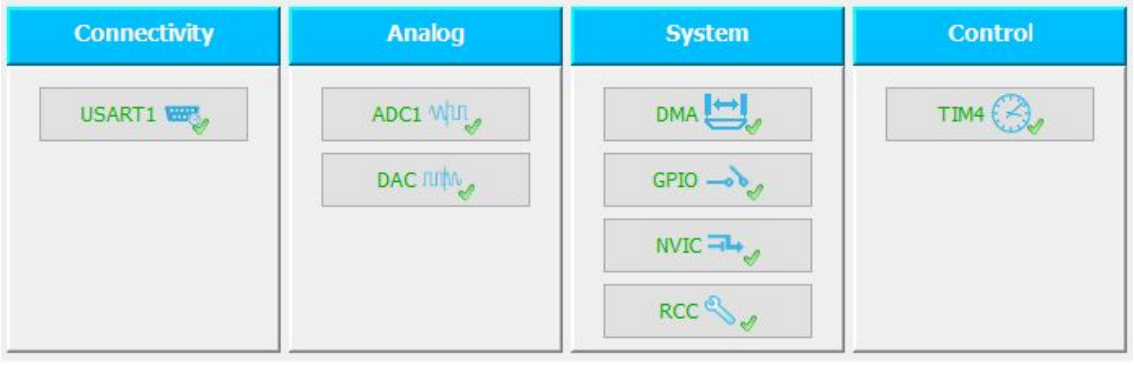
PCB Challenge - Documentation

Pinout

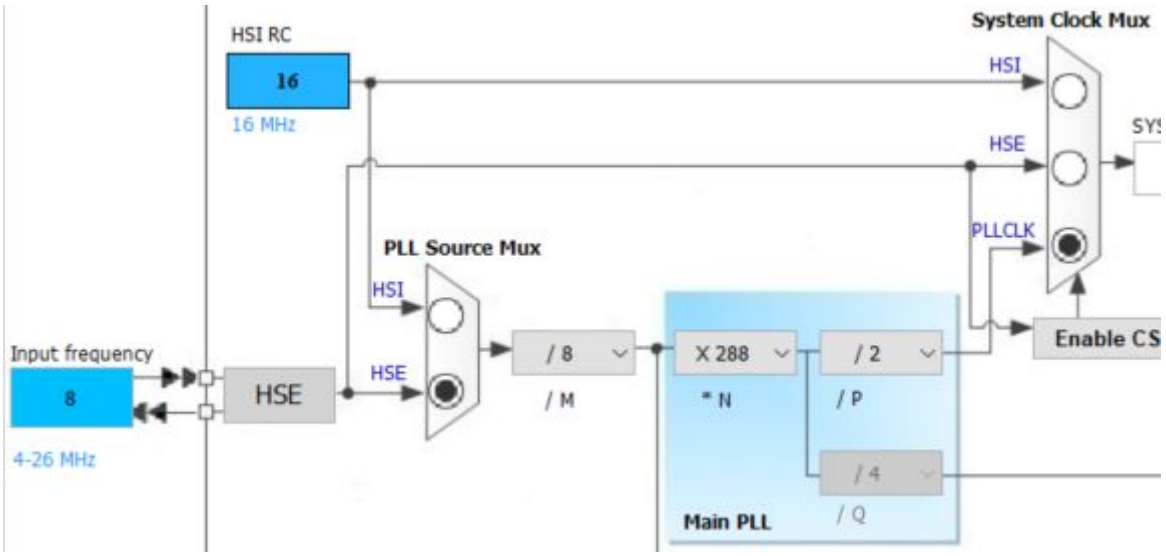


Main modules

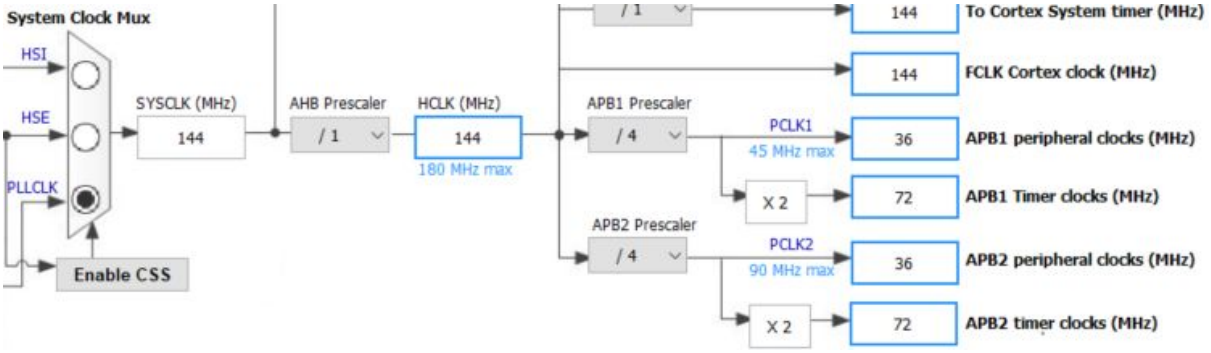


Clock configuration

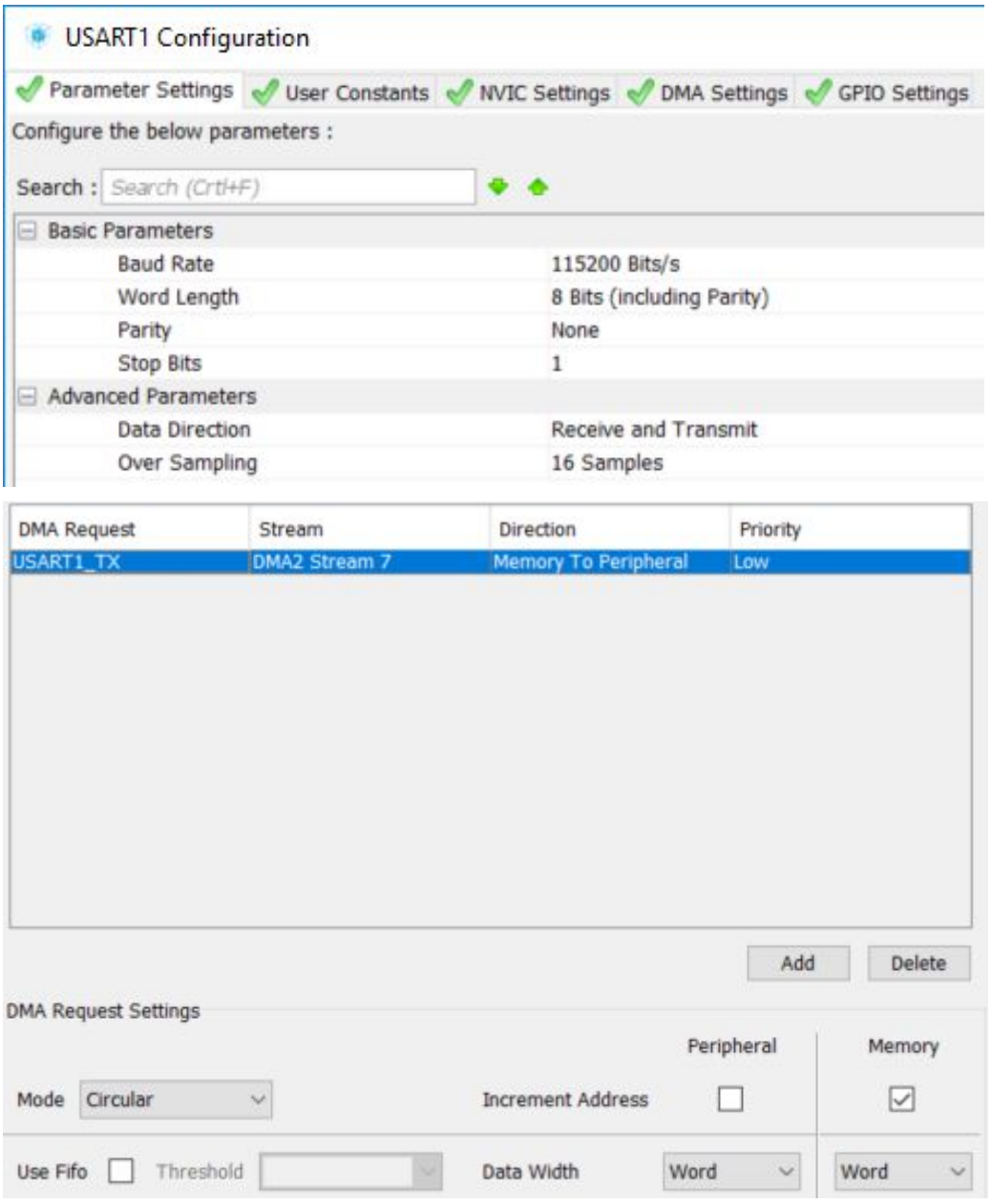
Left to the System Clock Mux:



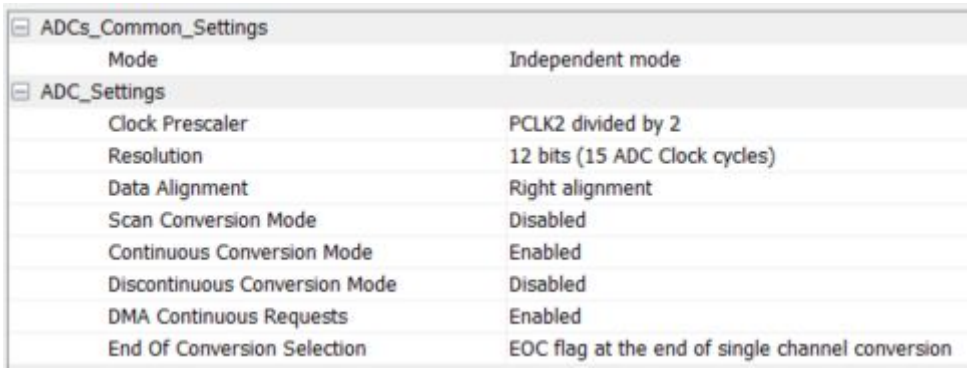
Right to the System Clock Mux:



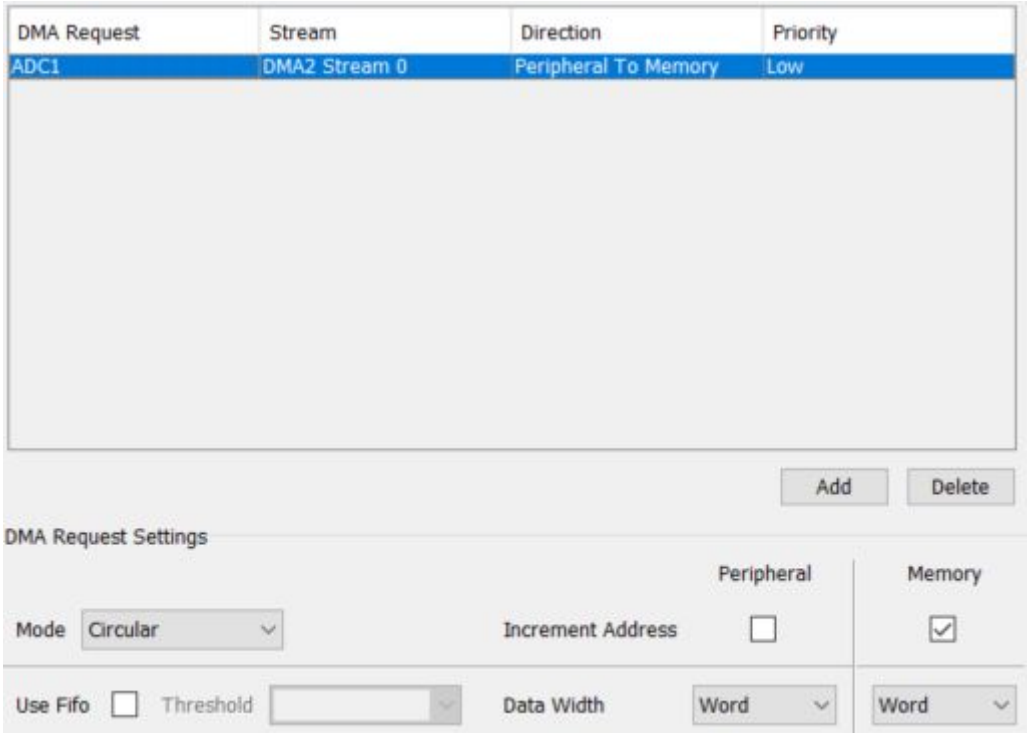
Configuration of the USART



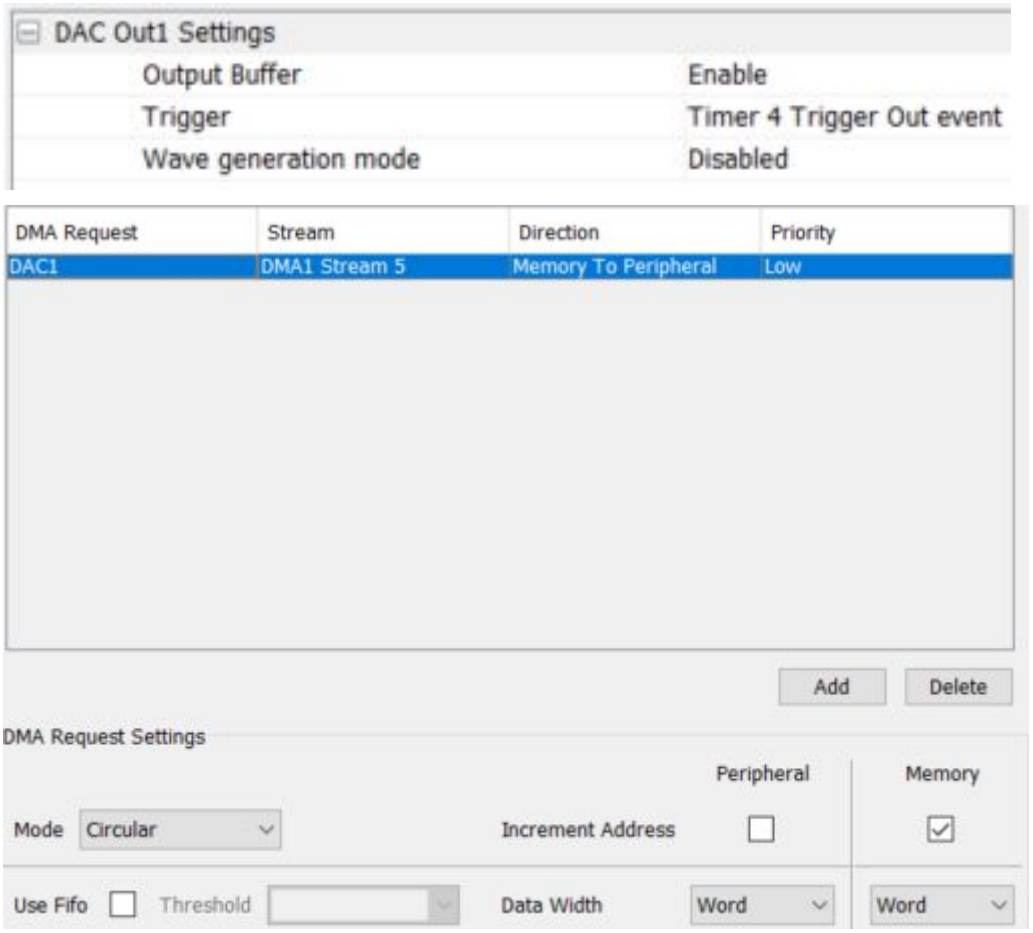
Configuration of the ADC



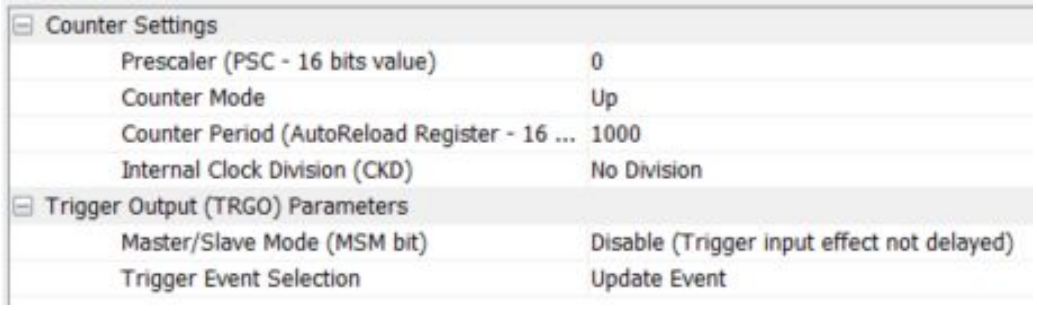
ADCs_Common_Settings	
Mode	Independent mode
ADC_Settings	
Clock Prescaler	PCLK2 divided by 2
Resolution	12 bits (15 ADC Clock cycles)
Data Alignment	Right alignment
Scan Conversion Mode	Disabled
Continuous Conversion Mode	Enabled
Discontinuous Conversion Mode	Disabled
DMA Continuous Requests	Enabled
End Of Conversion Selection	EOC flag at the end of single channel conversion



Configuration of the DAC



Configuration of Timer 4



Software

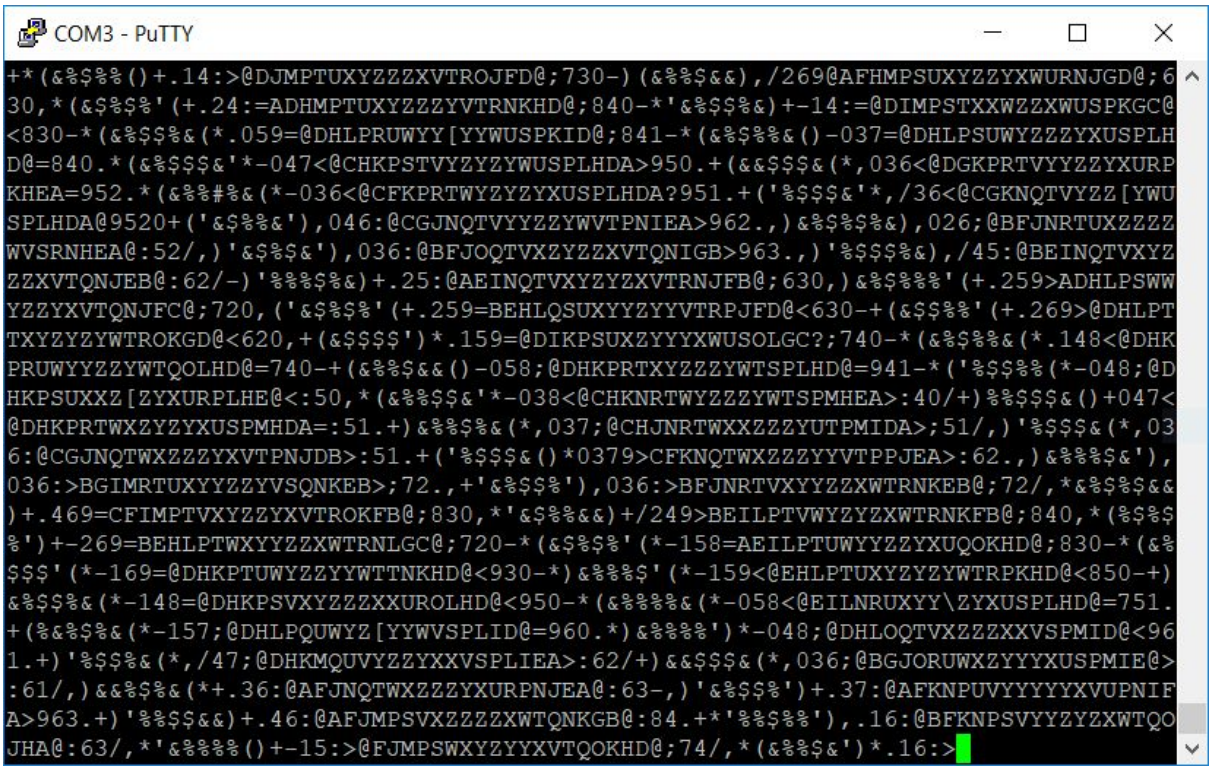
Some helpful variables

```
/* USER CODE BEGIN PV */
/* Private variables ---------------------------------------------------------*/
#define BUFFER_LENGTH 256 // Size of buffer arrays
// Parameters of the sine wave:
#define SIN_AMP 36
#define SIN_PERIOD 256
#define SIN_OFFSET 90
uint16_t i, j;
uint32_t buffer_input[BUFFER_LENGTH]; // Data to DAC
uint32_t buffer_output[BUFFER_LENGTH]; // Data from ADC
/* USER CODE END PV */
```

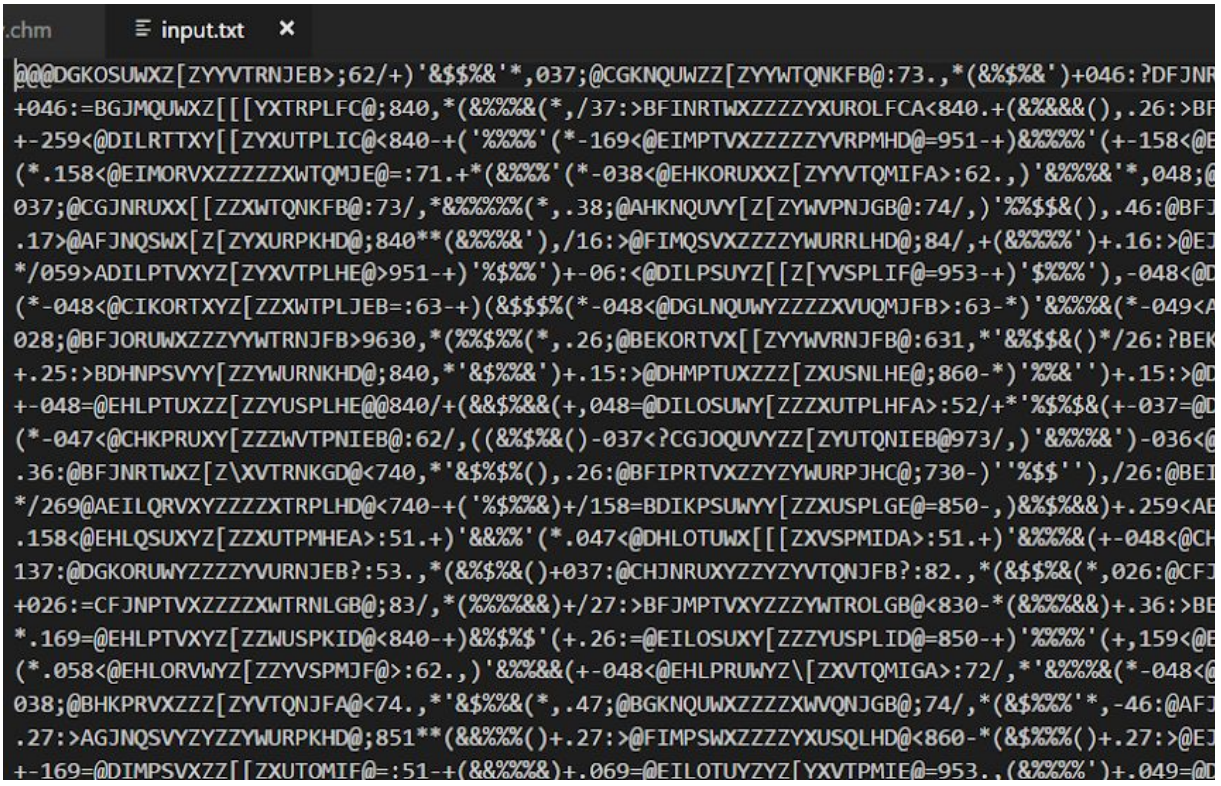
Main program

```
/* USER CODE BEGIN 2 */
for(i = 0; i < BUFFER_LENGTH; i++)
{
    buffer_input[i] = (int)(SIN_AMP * sin(2 * M_PI * i / SIN_PERIOD)) + SIN_OFFSET;
}
HAL_TIM_Base_Start(&htim4);
HAL_DAC_Start_DMA(&hdac, DAC_CHANNEL_1, (uint32_t *)buffer_input, BUFFER_LENGTH, DAC_ALIGN_12B_R);
HAL_ADC_Start_DMA(&hadc1, (uint32_t *)buffer_output, 32);
HAL_UART_Transmit_DMA(&huart1, (uint8_t *)buffer_output, 32);
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */

}
/* USER CODE END 3 */
```

Data stream on PuTTY



Data in a file before processing



C program to process the file

```c
int main(int argc, char * agrv[])
{
    char ch;
    int counter = 0;
    FILE * file_output = fopen("output.txt", "wt");
    FILE * file_input = fopen("input.txt", "rt");
    if (file_input == NULL)
    {
      perror("Error while opening the file.\n");
      exit(EXIT_FAILURE);
    }
    while((ch = fgetc(file_input)) != EOF)
    {
       counter++;
       if(counter < 1000)
       {
           fprintf(file_output, "%u\n", ch);
       }
    }
    printf("counter: %d\n", counter);
    fclose(file_input);
    return 0;
}
```

Sine wave (data from ADC) in Google Sheets

	Number of measurement	Data from ADC
2	1	58
3	2	63
4	3	68
5	4	70
6	5	74
7	6	78
8	7	82
9	8	85
10	9	87
11	10	89
12	11	90
13	12	91
14	13	91
15	14	90
16	15	88
17	16	87
18	17	84
19	18	81
20	19	79
21	20	74
22	21	70