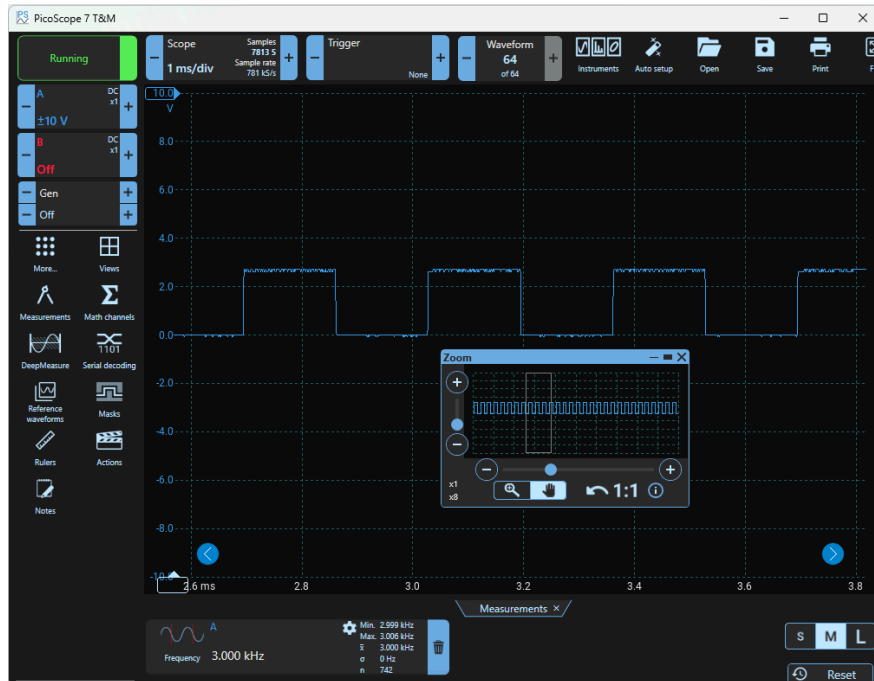


Regular conversions are enabled, with the external trigger conversion source set to Timer 2 Trigger Out event, with the trigger caused by a rising edge.

An external blue LED running as a GPIO\_output on pin PA11 is configured to toggle on ADC callback conversion, but due to the speed, its flickering is not visible to the human eye, and it is only used to show that callbacks are occurring, since it is set to switch off if the error handler is called. However, when viewing its signal on a scope shows an average frequency of 3000 Hz, meaning the ADC is triggered at twice that, 6000Hz.



Below, the code for the ADC callback can be seen:

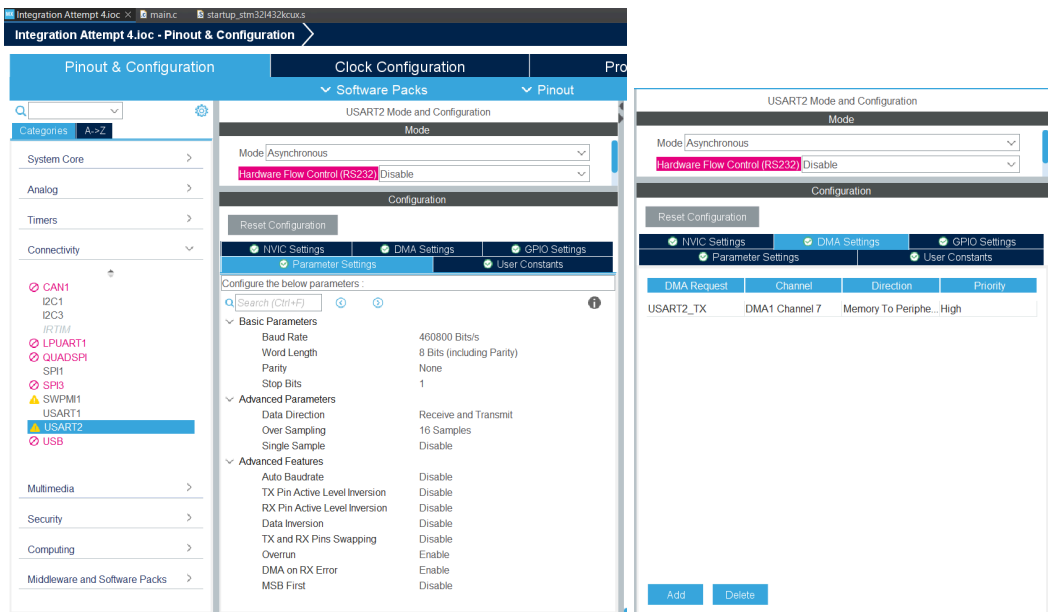
```
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef *hadc) {
    raw = HAL_ADC_GetValue(&hadc1);

    uint8_t *ADCValPtr = (uint8_t *)(&raw);
    int valLength = sizeof(raw);
    HAL_UART_Transmit_DMA(&huart2, ADCValPtr, valLength);

    HAL_GPIO_TogglePin(Blue_LED_GPIO_Port, Blue_LED_Pin);
}
```

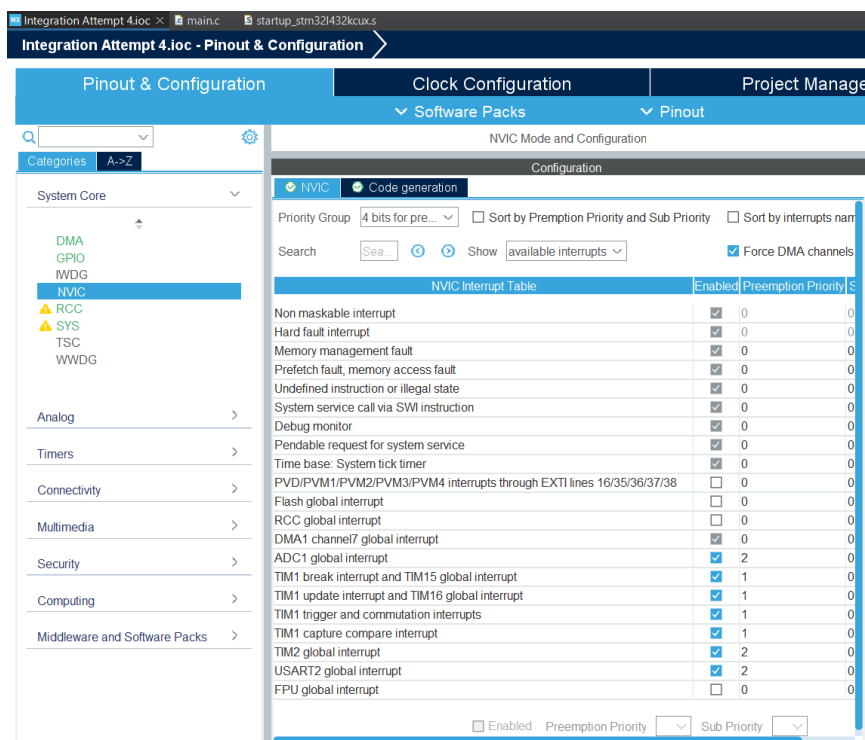
## UART Documentation

UART2 is used for serial communication with the CLI (as UART2 utilises the onboard ST-Link to communicate over the micro USB virtual COM port). Its baud rate is increased from the default 115200 bit/s to 460800 bit/s to allow for the highest data throughput possible while not compromising stability. A DMA request is set up on USART2\_TX with high priority in the memory to peripheral mode to get the DMA controller to transmit UART data instead of the CPU, freeing it for other tasks.



## NVIC Interrupt Documentation

Interrupt priorities are configured as shown, with many global interrupts enabled and preemption priorities set. Ultrasonic timing is given the highest priority (1) as the timing must be the most accurate/precise. ADC and UART are given secondary priority (2) as these are not as important.



## Ultrasonic Documentation

The ultrasonic sensor runs off of two interrupts, configured with **TIM1** and **TIM16**.

- ⚠ **TIM1**
- ⚠ **TIM2**
- TIM6**
- TIM7**
- ⚠ **TIM15**
- ⚠ **TIM16**

**TIM1** works as an Input Capture interrupt and is configured as follows:

TIM1 Mode and Configuration

Mode

Slave Mode

Disable

Trigger Source

Disable

Clock Source

Internal Clock

Channel1

Input Capture direct mode

Channel2

Disable

Channel3

Disable

Channel4

Disable

Channel5

Disable

Channel6

Disable

Combined Channels

Disable

Activate-Break-Input

Disable

Activate-Break-Input-2

Disable

Use ETR as Clearing Source

Disable

Configuration

Reset Configuration

✓ NVIC Settings

✓ DMA Settings

✓ GPIO Settings

✓ Parameter Settings

✓ User Constants

Configure the below parameters :

Search (Ctrl+F)

⌕

⌕

ⓘ

Counter Settings

Prescaler (PSC - 16 bits value)

31

Counter Mode

Up

Counter Period (AutoReload Regi...

65535

Internal Clock Division (CKD)

No Division

Repetition Counter (RCR - 16 bits...

0

auto-reload preload

Disable

Trigger Output (TRGO) Parameters

Master/Slave Mode (MSM bit)

Disable (Trigger input effect not delayed)

Trigger Event Selection TRGO

Reset (UG bit from TIMx\_EGR)

Trigger Event Selection TRGO2

Reset (UG bit from TIMx\_EGR)

Input Capture Channel 1

Polarity Selection

Both Edges

IC Selection

Direct

Prescaler Division Ratio

No division

Input Filter (4 bits value)

0

Configuration

Reset Configuration

✓ NVIC Settings	✓ DMA Settings	✓ GPIO Settings	
✓ Parameter Settings		✓ User Constants	
NVIC Interrupt Table		Enabled	Preemption Priority
TIM1 break interrupt and TIM15 global interrupt		✓	0
TIM1 update interrupt and TIM16 global interrupt		✓	0
TIM1 trigger and commutation interrupts		✓	0
TIM1 capture compare interrupt		✓	0

**TIM16** works as a timer interrupt and is configured as follows:

☒ Activated

Channel1 Disable

☐ Activate Break Input

☐ One Pulse Mode

Configuration

Reset Configuration

Parameter Settings User Constants NVIC Settings DMA Settings

Configure the below parameters :

Search (Ctrl+F)

Counter Settings

Prescaler (PSC - 16 bits value) 31

Counter Mode Up

Counter Period (AutoReload Regis... 65535

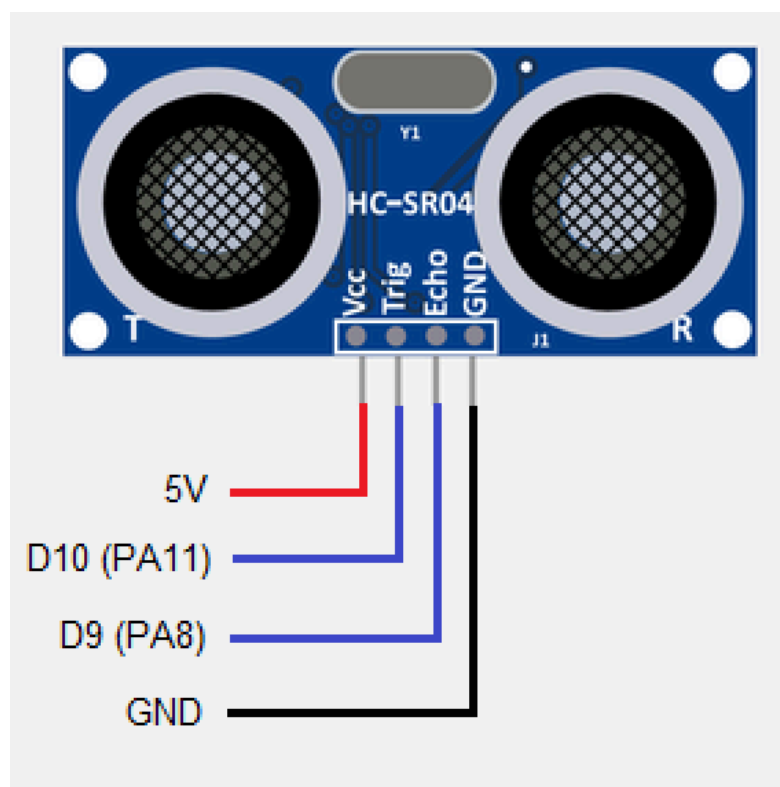
Internal Clock Division (CKD) No Division

Repetition Counter (RCR - 8 bits v... 0

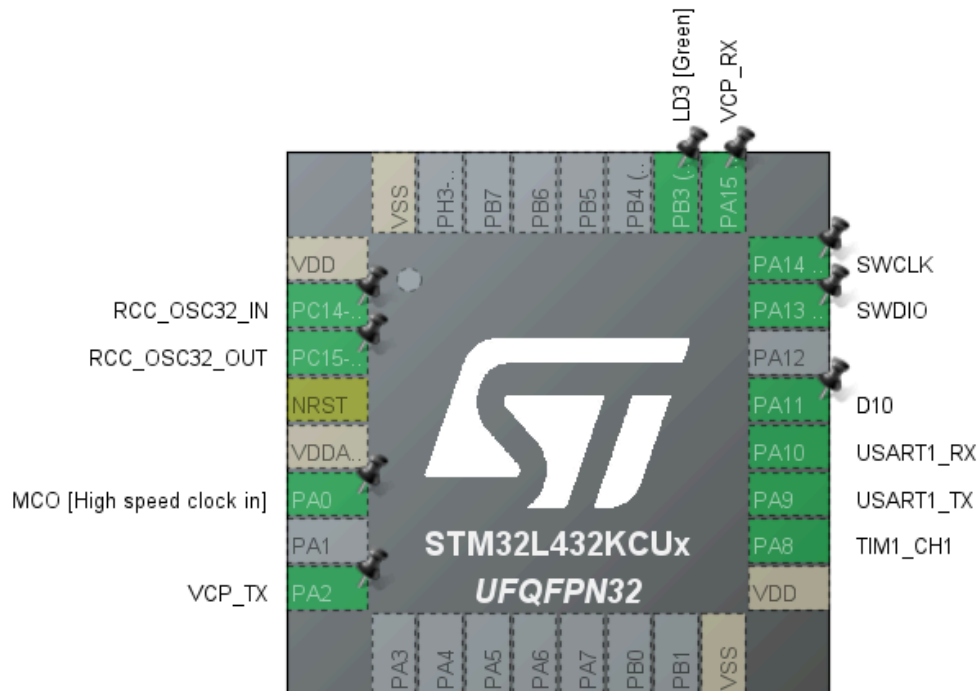
auto-reload preload Disable

Reset Configuration			
Parameter Settings	User Constants	NVIC Settings	DMA Settings
NVIC Interrupt Table			
	Enabled	Preemption Priority	Sub Priority
TIM1 update interrupt and TIM16 global interrupt	<input checked="" type="checkbox"/>	0	0

The MCU (STM32) interfaces with the Ultrasonic as described in the following diagram:



This is the config for the STM32 in CubeIDE (Note: D10 is a GPIO\_Output pin)



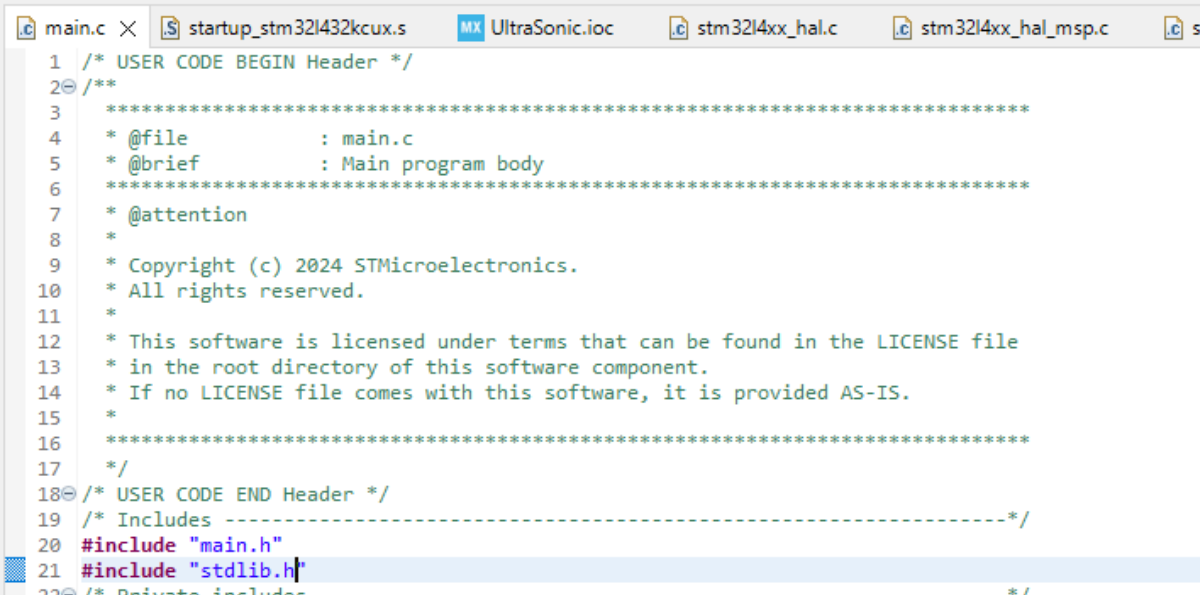
The callback function for **TIM1** is as follows:

```

64  /* Private user code -----*/
65  /* USER CODE BEGIN 0 */
66
67  int Is_First_Captured = 0;
68  uint16_t timer2 = 0;
69  uint16_t timer1 = 0;
70
71  uint16_t usTime = 0;
72
73  char flag = 'S';
74  uint8_t * flagpointer = (uint8_t *) &flag;
75
76  void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim) {
77
78      if (Is_First_Captured==0)
79      {
80          timer1 = (uint16_t) HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_1);
81          Is_First_Captured = 1;
82      }
83
84      else
85      {
86
87          timer2 = (uint16_t) HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_1);
88
89          usTime = abs(timer1 - timer2);
90
91          if (usTime > 20000){
92              return;
93          }
94          uint8_t * datapointer = (uint8_t *) &usTime;
95
96          HAL_UART_Transmit(&huart2, flagpointer, 1, 100);
97          HAL_UART_Transmit(&huart2, datapointer, 2, 100);
98
99
100         __HAL_TIM_SET_COUNTER(htim, 0); // reset the counter
101         Is_First_Captured = 0; // set back to false
102
103     }
104 }
105

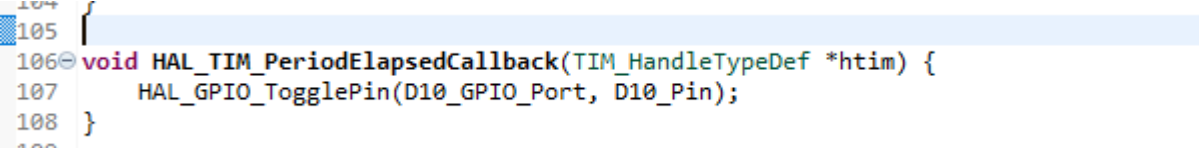
```

Note: the function “abs” is a stdlib function, you **must include stdlib.h** under the “Includes” header as follows:



```
1  /* USER CODE BEGIN Header */
2  /**
3   *
4   * @file          : main.c
5   * @brief         : Main program body
6   *
7   * @attention
8   *
9   * Copyright (c) 2024 STMicroelectronics.
10  * All rights reserved.
11  *
12  * This software is licensed under terms that can be found in the LICENSE file
13  * in the root directory of this software component.
14  * If no LICENSE file comes with this software, it is provided AS-IS.
15  *
16  *
17  */
18 /* USER CODE END Header */
19 /* Includes -----*/
20 #include "main.h"
21 #include "stdlib.h"
22 /* Private includes -----*/
```

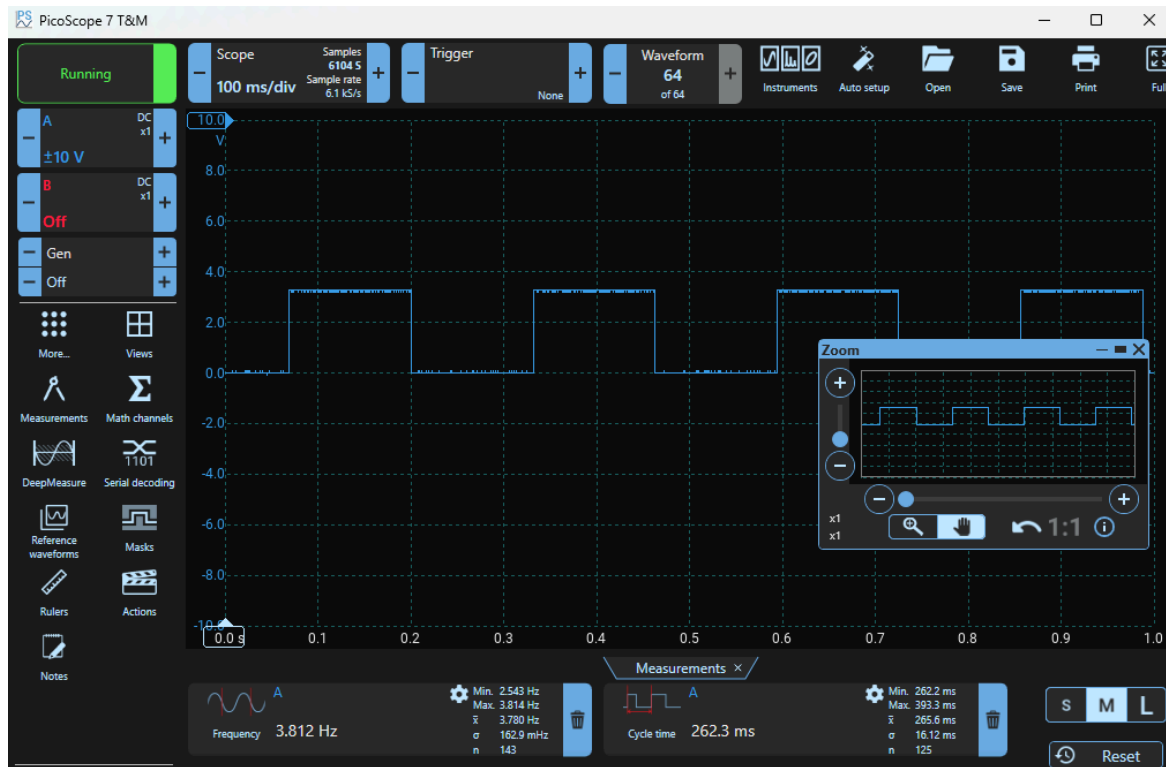
The callback function for **TIM16** (Timer interrupt) is as follows:



```
105 {
106 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
107     HAL_GPIO_TogglePin(D10_GPIO_Port, D10_Pin);
108 }
109
```

### Explanation:

**TIM16** interrupts every time it counts over 65535 microseconds (the counter period), and toggles D10 (the trigger pin) in the callback function. This means that every ~130 milliseconds the trigger pin goes high and a sonic burst is sent from the Ultrasonic Sensor. This can be seen, as onboard LD3 is toggled every time the US callback function is triggered. When connected to a scope we see the following:



Here, the cycle time is measured at 262ms, which is the time taken for two US measurements to be made since the LD3 pin is toggling.

**TIM1** interrupts every time D9 (the echo pin) has an edge, i.e. every time D9 goes from low to high or high to low. So, to measure the amount of time between each edge, `HAL_TIM_ReadCapturedValue` is stored at each edge (timer1 and timer2), and the difference between them is taken (`usTime`).

There is a chance that the US does not receive a signal back, in which case the value of `usTime` will be ~65000. To handle this error in the IC function we check if `usTime` is greater than 20,000, and if it is the function does not transmit the value.

\* 20,000 is greater than what the US could receive at its max range. This is why the prescaler value of 32-1 was chosen, we are triggering the US sensor at the maximum frequency possible to allow for the echo to be received at max distance before a new trigger is sent.