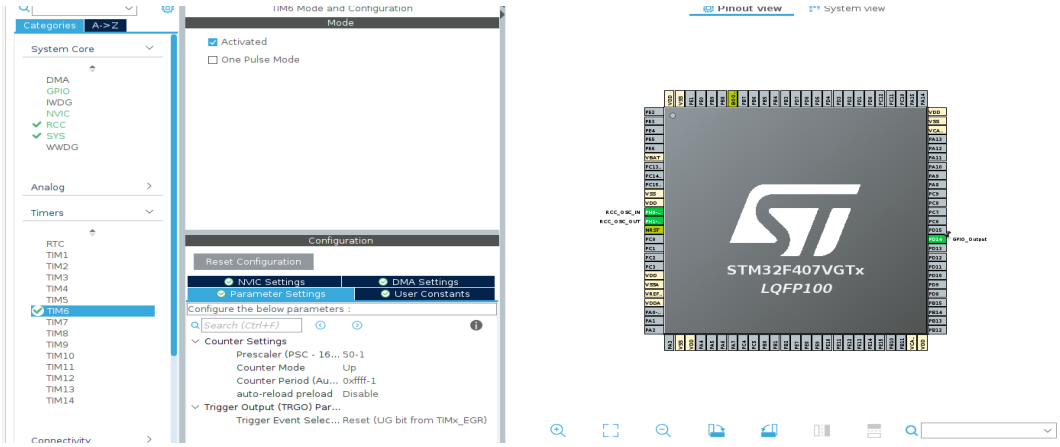
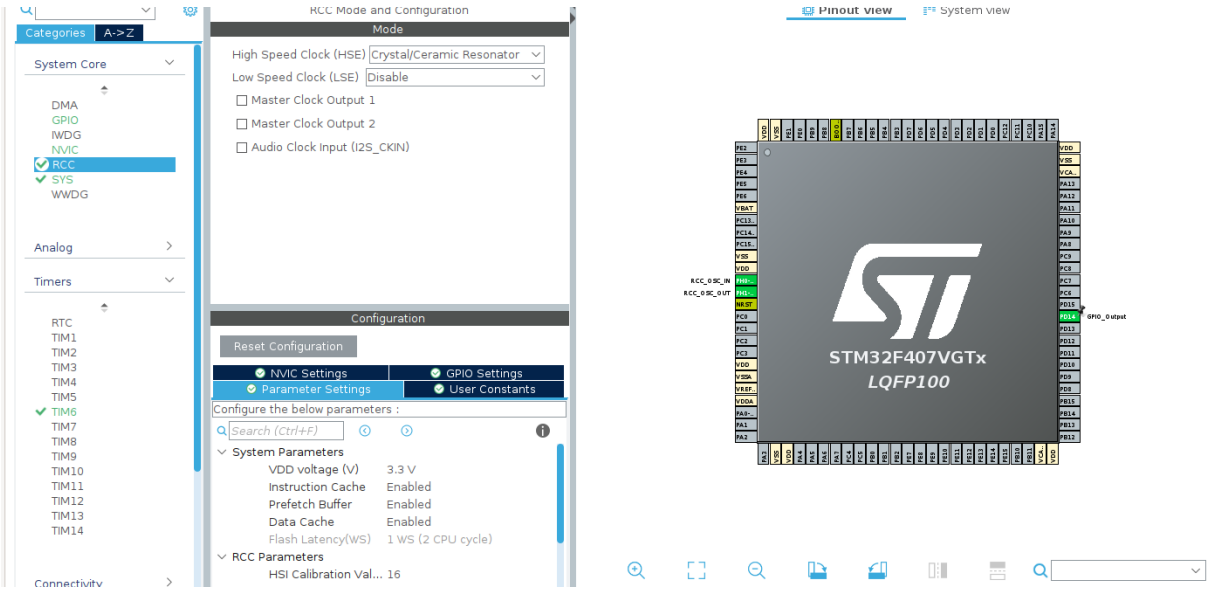
Send temprature and humidity data on stm32 using DHT-22

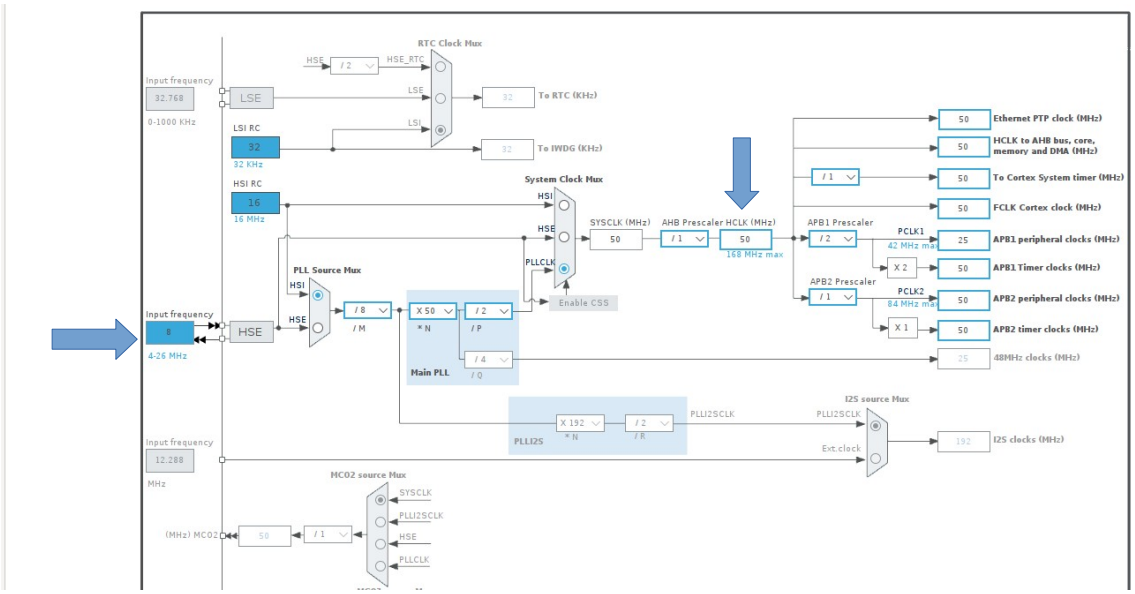
configure the stm32 cube ide as below step:
1. pd14 as GPIO_Output
2. configure timer tim6 prescalor (50 – 1) ,Counter period(0xffff-1),other will not change



3. configure RCC as below



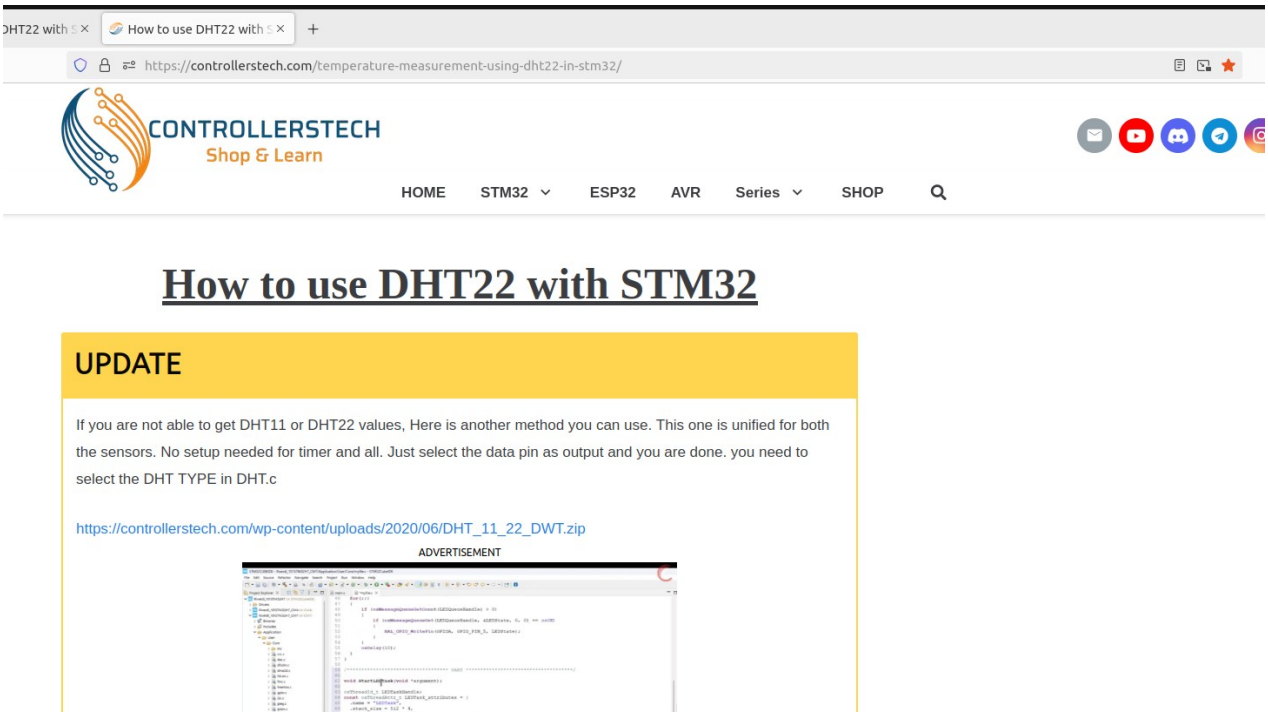
4. Clock configuration



5. This is link for main.c code we visit this side

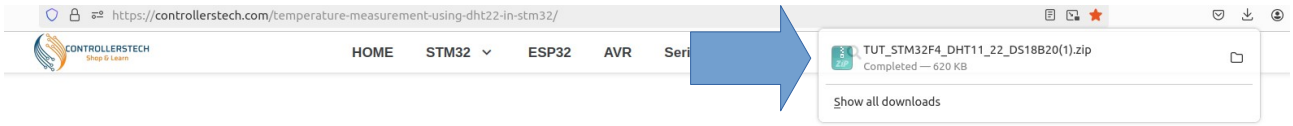
https://controllerstech.com/temperature-measurement-using-dht22-in-stm32

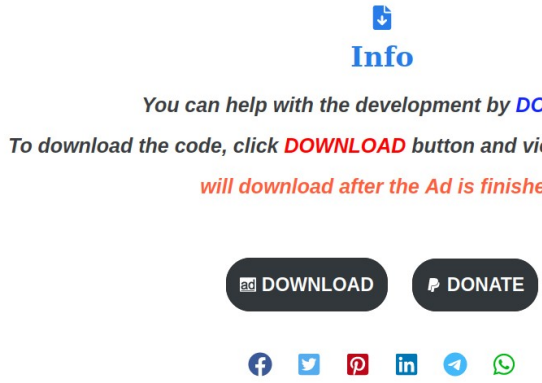
6. on this site scroll down



7.Download the zip folder for main.c code

7. Note Downloaded file name





7.According to the main file of this code we will change GPIO

```
/* USER CODE BEGIN Header */
/**
  ******************************************************************************
  * @file           : main.c
  * @brief          : Main program body
  ******************************************************************************
  * @attention
  *
  * Copyright (c) 2024 STMicroelectronics.
  * All rights reserved.
  *
  * This software is licensed under terms that can be found in the LICENSE file
  * in the root directory of this software component.
  * If no LICENSE file comes with this software, it is provided AS-IS.
  *
```

```
  ******************************************************************************
  */
/* USER CODE END Header */
/* Includes ------------------------------------------------------------------*/
#include "main.h"

/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */
#include "stdio.h"
/* USER CODE END Includes */

/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define ------------------------------------------------------------*/
/* USER CODE BEGIN PD */

/* USER CODE END PD */

/* Private macro -------------------------------------------------------------*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables ---------------------------------------------------------*/
TIM_HandleTypeDef htim6;

/* USER CODE BEGIN PV */

/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_TIM6_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */
void delay (uint16_t time)
{
	/* change your code here for the delay in microseconds */
	__HAL_TIM_SET_COUNTER(&htim6, 0);
	while ((__HAL_TIM_GET_COUNTER(&htim6))<time);
}
uint8_t Rh_byte1, Rh_byte2, Temp_byte1, Temp_byte2;
uint16_t SUM, RH, TEMP;

float Temperature = 0;
float Humidity = 0;
uint8_t Presence = 0;

void Set_Pin_Output (GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin)
{
	GPIO_InitTypeDef GPIO_InitStruct = {0};
	GPIO_InitStruct.Pin = GPIO_Pin;
	GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
	GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
```

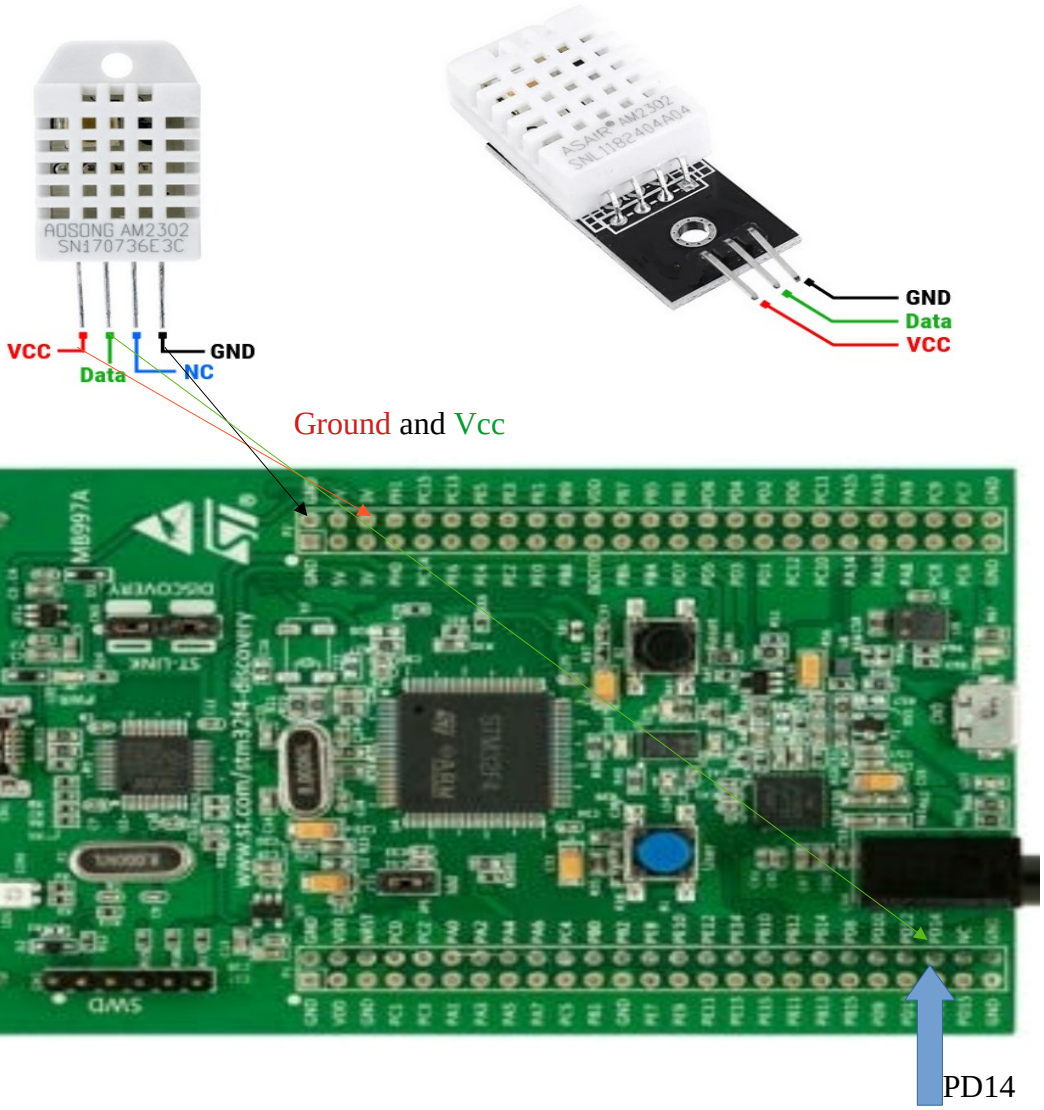
```
	HAL_GPIO_Init(GPIOx, &GPIO_InitStruct);
}

void Set_Pin_Input (GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin)
{
	GPIO_InitTypeDef GPIO_InitStruct = {0};
	GPIO_InitStruct.Pin = GPIO_Pin;
	GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
	GPIO_InitStruct.Pull = GPIO_PULLUP;
	HAL_GPIO_Init(GPIOx, &GPIO_InitStruct);
}
#define DHT22_PORT GPIOD
#define DHT22_PIN GPIO_PIN_14

void DHT22_Start (void)
{
	Set_Pin_Output(DHT22_PORT, DHT22_PIN); // set the pin as output
	HAL_GPIO_WritePin (DHT22_PORT, DHT22_PIN, 0);   // pull the pin low
	delay(1200);   // wait for > 1ms

	HAL_GPIO_WritePin (DHT22_PORT, DHT22_PIN, 1);   // pull the pin high
	delay (20);   // wait for 30us

	Set_Pin_Input(DHT22_PORT, DHT22_PIN);   // set as input
}

uint8_t DHT22_Check_Response (void)
{
	Set_Pin_Input(DHT22_PORT, DHT22_PIN);   // set as input
	uint8_t Response = 0;
	delay (40);  // wait for 40us
	if (!(HAL_GPIO_ReadPin (DHT22_PORT, DHT22_PIN))) // if the pin is low
	{
		delay (80);   // wait for 80us

		if ((HAL_GPIO_ReadPin (DHT22_PORT, DHT22_PIN))) Response = 1;  // if
the pin is high, response is ok
		else Response = -1;
	}

	while ((HAL_GPIO_ReadPin (DHT22_PORT, DHT22_PIN)));   // wait for the pin
to go low
	return Response;
}

uint8_t DHT22_Read (void)
{
	uint8_t i,j;
	for (j=0;j<8;j++)
	{
		while (!(HAL_GPIO_ReadPin (DHT22_PORT, DHT22_PIN)));   // wait for
the pin to go high
		delay (40);   // wait for 40 us

		if (!(HAL_GPIO_ReadPin (DHT22_PORT, DHT22_PIN)))   // if the pin is
low
		{
			i&= ~(1<<(7-j));   // write 0
		}
		else i|= (1<<(7-j));  // if the pin is high, write 1
		while ((HAL_GPIO_ReadPin (DHT22_PORT, DHT22_PIN)));  // wait for the
pin to go low
```

```
      }

      return i;
}


/* USER CODE END 0 */

/**
  * @brief  The application entry point.
  * @retval int
  */
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */

  /* MCU Configuration--------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick.
*/
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_TIM6_Init();
  /* USER CODE BEGIN 2 */
  HAL_TIM_Base_Start(&htim6);
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
        DHT22_Start();
              Presence = DHT22_Check_Response();
              Rh_byte1 = DHT22_Read ();
              Rh_byte2 = DHT22_Read ();
              Temp_byte1 = DHT22_Read ();
              Temp_byte2 = DHT22_Read ();
              SUM = DHT22_Read();

              TEMP = ((Temp_byte1<<8)|Temp_byte2);
              RH = ((Rh_byte1<<8)|Rh_byte2);

              Temperature = (float) (TEMP/10.0);
              Humidity = (float) (RH/10.0);

     // HAL_Delay(1000);
```

```c
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
}

/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

  /** Configure the main internal regulator output voltage
  */
  __HAL_RCC_PWR_CLK_ENABLE();
  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

  /** Initializes the RCC Oscillators according to the specified parameters
  * in the RCC_OscInitTypeDef structure.
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
  RCC_OscInitStruct.HSIState = RCC_HSI_ON;
  RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
  RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
  RCC_OscInitStruct.PLL.PLLM = 8;
  RCC_OscInitStruct.PLL.PLLN = 50;
  RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
  RCC_OscInitStruct.PLL.PLLQ = 4;
  if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    Error_Handler();
  }

  /** Initializes the CPU, AHB and APB buses clocks
  */
  RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                              |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
  RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

  if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_1) != HAL_OK)
  {
    Error_Handler();
  }
}

/**
  * @brief TIM6 Initialization Function
  * @param None
  * @retval None
  */
static void MX_TIM6_Init(void)
{

  /* USER CODE BEGIN TIM6_Init 0 */
```

```c
  /* USER CODE END TIM6_Init 0 */

  TIM_MasterConfigTypeDef sMasterConfig = {0};

  /* USER CODE BEGIN TIM6_Init 1 */

  /* USER CODE END TIM6_Init 1 */
  htim6.Instance = TIM6;
  htim6.Init.Prescaler = 50-1;
  htim6.Init.CounterMode = TIM_COUNTERMODE_UP;
  htim6.Init.Period = 0xffff-1;
  htim6.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
  if (HAL_TIM_Base_Init(&htim6) != HAL_OK)
  {
    Error_Handler();
  }
  sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
  sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
  if (HAL_TIMEx_MasterConfigSynchronization(&htim6, &sMasterConfig) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN TIM6_Init 2 */

  /* USER CODE END TIM6_Init 2 */

}

/**
  * @brief GPIO Initialization Function
  * @param None
  * @retval None
  */
static void MX_GPIO_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};
/* USER CODE BEGIN MX_GPIO_Init_1 */
/* USER CODE END MX_GPIO_Init_1 */

  /* GPIO Ports Clock Enable */
  __HAL_RCC_GPIOH_CLK_ENABLE();
  __HAL_RCC_GPIOD_CLK_ENABLE();

  /*Configure GPIO pin Output Level */
  HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);

  /*Configure GPIO pin : PD14 */
  GPIO_InitStruct.Pin = GPIO_PIN_14;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);

/* USER CODE BEGIN MX_GPIO_Init_2 */
/* USER CODE END MX_GPIO_Init_2 */
}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */
```

```c
/**
  * @brief  This function is executed in case of error occurrence.
  * @retval None
  */
void Error_Handler(void)
{
  /* USER CODE BEGIN Error_Handler_Debug */
  /* User can add his own implementation to report the HAL error return state */
  __disable_irq();
  while (1)
  {
  }
  /* USER CODE END Error_Handler_Debug */
}

#ifdef  USE_FULL_ASSERT
/**
  * @brief  Reports the name of the source file and the source line number
  *         where the assert_param error has occurred.
  * @param  file: pointer to the source file name
  * @param  line: assert_param error line source number
  * @retval None
  */
void assert_failed(uint8_t *file, uint32_t line)
{
  /* USER CODE BEGIN 6 */
  /* User can add his own implementation to report the file name and line
number,
     ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
  /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */
```


AOSONG AM2302
SN170736E3C
VCC
Data
NC
GND
ASAIR AM2302
SN:1182404A04
GND
Data
VCC
Ground and Vcc
PD14