

**DATE: 16/9/2024**

**NAME: LATHIKA L J**

**ROLL NO :212223220050**

**DEPARTMENT: IT**

## **EXPERIMENT--04-INTERFACING IOT DEVELOPMENT BOARD AND CONFIGURE USART FOR TRANSFERRING STRINGS**

**Aim: To Interface iot development board for configuring the the usart and transfer strings though it**

**Components required: STM32 CUBE IDE, ARM IOT development board, STM programmer tool, Serial port utility tool**

### **Theory**

The full form of an ARM is an advanced reduced instruction set computer (RISC) machine, and it is a 32-bit processor architecture expanded by ARM holdings. The applications of an ARM processor include several microcontrollers as well as processors. The architecture of an ARM processor was licensed by many corporations for designing ARM processor-based SoC products and CPUs. This allows the corporations to manufacture their products using ARM architecture. Likewise, all main semiconductor companies will make ARM-based SOC's such as Samsung, Atmel, TI etc.

# 1.select the appropriate pins as gipo, in or out, USART or required options and configure

The screenshot shows the STM32CubeIDE interface. The 'Pinout & Configuration' window is open, displaying a list of categories on the left and a pinout diagram on the right. The pinout diagram shows the PA5 pin selected, with a dropdown menu showing various functions including Reset\_State, ADC1\_IN5, CEC, DAC1\_OUT2, I2S1\_CK, LPTIM2\_ETR, SPI1\_SCK, TIM2\_CH1, TIM2\_ETR, UCPD1\_FRSTX1, UCPD1\_FRSTX2, USART3\_TX, GPIO\_Input, GPIO\_Output, GPIO\_Analog, EVENTOUT, and GPIO\_EXTI5. The console at the bottom shows 'No consoles to display at this time.' An 'Updates Available' notification is also present in the bottom right corner.

workspace\_1.8.0 - Device Configuration Tool - STM32CubeIDE

File Edit Navigate Search Project Run Window Help

Project Explorer

- 123
- adc
- DEMO
- exp-01
- EXP-02
- EXP1
- exp4
  - Includes
  - Core
  - Drivers
    - exp4.ioc
    - STM32G071RBTX\_FLASH.ld
  - interrupt
  - Interruptconcept
  - irsensor exp2
  - IR Sensor Interfacing\_Digital Read (in IR\_GPIO\_READ)
- LEDONOFF
- LoRaWAN\_End\_Node (in STM32CubeIDE)
- LQBOARDUSART (in LQBOARD)
- lux\_sensor
- proj1
- PUSHBUTTON
- Soil moisture sensor
- ssssss

main.c exp4.ioc

Pinout & Configuration Clock Configuration Project Manager Tools

Software Packs Pinout

Pinout view System view

Categories A-Z

- System Core
- Analog
- Timers
- Connectivity
- Multimedia
- Computing
- Middleware
- Utilities

PA5

- Reset\_State
- ADC1\_IN5
- CEC
- DAC1\_OUT2
- I2S1\_CK
- LPTIM2\_ETR
- SPI1\_SCK
- TIM2\_CH1
- TIM2\_ETR
- UCPD1\_FRSTX1
- UCPD1\_FRSTX2
- USART3\_TX
- GPIO\_Input
- GPIO\_Output
- GPIO\_Analog
- EVENTOUT
- GPIO\_EXTI5

Console

No consoles to display at this time.

Updates Available

Updates are available for your software. Click to review and install updates. You will be reminded in 4 Hours. Set reminder [preferences](#)

The screenshot displays the STM32CubeIDE interface for configuring a pin. The 'Pinout & Configuration' tab is active, showing the 'GPIO Mode and Configuration' for pin PA5. The pin is configured as an output with push-pull mode, no pull-up or pull-down, and a low output level. The maximum output speed is set to Low. The user label is empty.

**GPIO Mode and Configuration**

Pin No.	Signal	GP...	GPIO ...	GPIO ...	Maxim...	Fast M...	User L...	Modified
PA5	n/a	Low	Output...	No pull...	Low	n/a		

**PA5 Configuration :**

- GPIO output level: Low
- GPIO mode: Output Push Pull
- GPIO Pull-up/Pull-down: No pull-up and no pull-down
- Maximum output speed: Low
- User Label:

The right side of the interface shows a pinout diagram of the STM32G071RBTx LQFP64 package. Pin PA5 is highlighted in green and labeled 'GPIO\_Output'.

**Updates Available**

Updates are available for your software. Click to review and install updates. You will be reminded in 4 Hours. Set reminder [preferences](#)

configure in the usart 2 as asynchronous mode and set the baud rate as 115200 as shown below

The screenshot displays the STM32CubeIDE interface for configuring a USART2 peripheral. The main window is titled "workspace\_1.8.0 - Device Configuration Tool - STM32CubeIDE". The left sidebar shows the Project Explorer with files like `adc`, `exp5keypad.ioc`, and `main.c`. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The main workspace is divided into several panes:

- Pinout & Configuration**: This pane is active and shows the "USART2 Mode and Configuration" settings. The Mode is set to "Asynchronous". The "Hardware Flow Control (RS232)" is set to "Disable". The "Slave Select(NSS) Management" is set to "Disable". A tooltip indicates: "Slave Select(NSS) Management Disabled. Active only with Synchronous Slave Mode".
- Configuration**: This pane shows the "Parameter Settings" for the USART2 peripheral. The Baud Rate is set to 115200 Bits/s, Word Length is 8 Bits (including Parity), and Parity is None.
- Pinout view**: This pane shows the pinout of the STM32 microcontroller. The pins are color-coded: green for digital, blue for analog, and yellow for power. The USART2 pins are highlighted in green.

The bottom status bar shows the Build Analyzer and Static Stack Analyzer. The Build Analyzer displays the memory regions and details for the `adc.elf` file:

Region	Start address	End address	Size	Free	Used	Usage (%)
RAM	0x20000000	0x20010000	64 KB	62.22 KB	1.78 KB	2.78%
RAM2	0x10000000	0x10008000	32 KB	32 KB	0 B	0.00%
FLASH	0x08000000	0x08040000	256 KB	242.63 KB	13.37 KB	5.22%

2.click on cntrl+S , automaticall C program will be generated

The screenshot displays the STM32CubeIDE interface for configuring the STM32G071RBTx LQFP64 microcontroller. The 'Pinout & Configuration' tab is active, showing the 'GPIO Mode and Configuration' for the PA5 pin. The configuration settings are as follows:

Pin No.	Signal	GP...	GPIO...	Maxim...	Fast M...	User L...	Modified
PA5	n/a	Low	Output...	No pull...	Low	n/a	<input type="checkbox"/>

The 'PA5 Configuration' section shows the following settings:

- GPIO output level: Low
- GPIO mode: Output Push Pull
- GPIO Pull-up/Pull-down: No pull-up and no pull-down
- Maximum output speed: Low
- User Label: (empty)

The pinout diagram on the right shows the physical layout of the 64 pins, with PA5 labeled as 'GPIO Output'.

**Updates Available**

Updates are available for your software. Click to review and install updates. You will be reminded in 4 Hours. Set reminder [preferences](#)



The screenshot displays the STM32CubeIDE environment. The main editor window shows the `main.c` file with the following code structure:

```

1  /* USER CODE BEGIN Header */
2  /**
3   *
4   * @file      : main.c
5   * @brief     : Main program body
6   *
7   * @attention
8   *
9   * Copyright (c) 2023 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
22 /* Private includes */
23 /* USER CODE BEGIN Includes */
24
25 /* USER CODE END Includes */
26
27 /* Private typedef */
28 /* USER CODE BEGIN PTD */
29
30 /* USER CODE END PTD */
31
32 /* Private define */
33 /* USER CODE BEGIN PD */
34 /* USER CODE END PD */
35
36 /* Private macro */
37 /* USER CODE BEGIN PM */
38
39 /* USER CODE END PM */
40
41 /* Private variables */
42
43 /* USER CODE BEGIN PV */
44
45 /* USER CODE END PV */
46
47 /* Private function prototypes */
48 void SystemClock_Config(void);
49 /* USER CODE BEGIN PFP */
50
51 /* USER CODE END PFP */
52
53 /* Private user code */

```

The Project Explorer on the left shows the project structure, including files like `main.c`, `exp4.ioc`, and various driver files. The Outline view on the right lists the functions defined in `main.h`, such as `SystemClock_Config(void)`, `main(void)`, and `Error_Handler(void)`.

A 'Progress Information' dialog box is overlaid on the editor, indicating that the 'Device Configuration Tool' is updating the code. The dialog includes a green progress bar and a 'Cancel' button.

The bottom status bar shows the current state of the IDE, including 'Writable', 'Smart Insert', and '1:1:0' zoom level. The system tray at the bottom right shows the date and time as '19-03-2023' and '21:48'.

8. edit the program and as per required

workspace\_1.8.0 - exp4/Core/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- 123
- adc
- DEMO
- exp-01
- EXP-02
- EXP1
- exp4
  - Includes
  - Core
  - Drivers
    - exp4.ioc
    - STM32G071R8TX\_FLASH.ld
  - interrupt
  - Interruptconcept
  - irsensor exp2
    - IR Sensor Interfacing\_Digital Read (in IR\_GPIO\_READ)
  - LEDONOFF
    - LoRaWAN\_End\_Node (in STM32CubeIDE)
  - LQBOARDUSART (in LQBOARD)
  - lux\_sensor
  - proj1
  - PUSHBUTTON
    - Soil moisture sensor
  - ssssss

main.c

```

1  /* USER CODE BEGIN Header */
2  /**
3   *
4   * @file          : main.c
5   * @brief         : Main program body
6   *
7   * @attention
8   *
9   * Copyright (c) 2023 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  /* USER CODE END Header */
18  /* Includes */
19  #include "main.h"
20
21  /* Private includes */
22  /* USER CODE BEGIN Includes */
23
24  /* USER CODE END Includes */
25
26  /* Private typedef */
27  /* USER CODE BEGIN PTD */
28
29  /* USER CODE END PTD */
30
31  /* Private define */
32  /* USER CODE BEGIN PD */
33
34  /* USER CODE END PD */
35
36  /* Private macro */
37  /* USER CODE BEGIN PM */
38
39  /* USER CODE END PM */
40
41  /* Private variables */
42
43  /* USER CODE BEGIN PV */
44
45  /* USER CODE END PV */
46
47  /* Private function prototypes */
48  void SystemClock_Config(void);
49  static void MX_GPIO_Init(void);
50  /* USER CODE BEGIN PFP */
51
52  /* USER CODE END PFP */
53

```

Outline

- main.h
- SystemClock\_Config(void) : void
- MX\_GPIO\_Init(void) : void
- main(void) : int
- SystemClock\_Config(void) : void
- MX\_GPIO\_Init(void) : void
- Error\_Handler(void) : void
- assert\_failed(uint8\_t\*, uint32\_t) : void

Problems Tasks Console Properties

No consoles to display at this time.

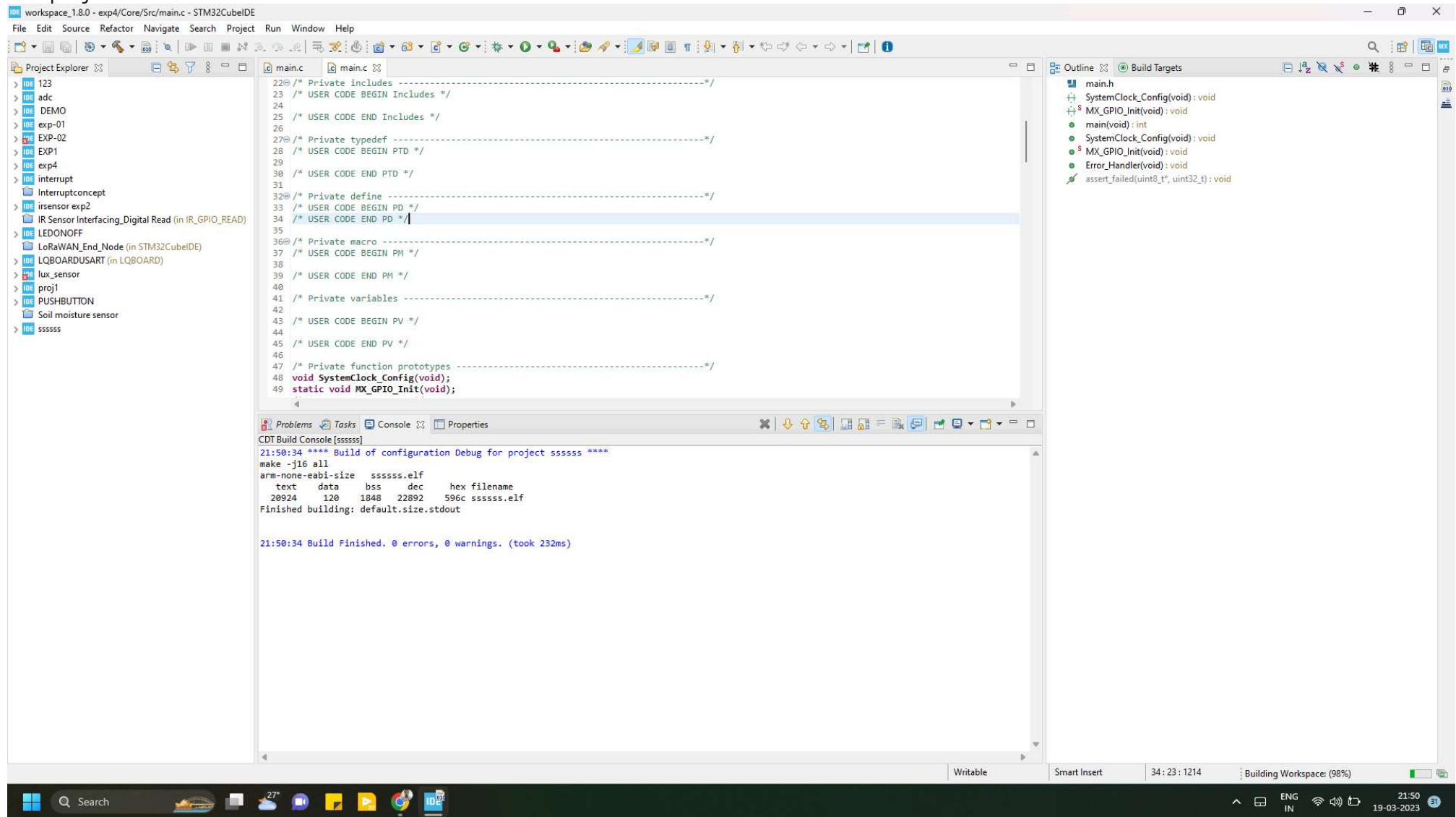
Writable Smart Insert 1:1:0

Updates Available

Updates are available for your software. Click to review and install updates. You will be reminded in 4 Hours. Set reminder [preferences](#)

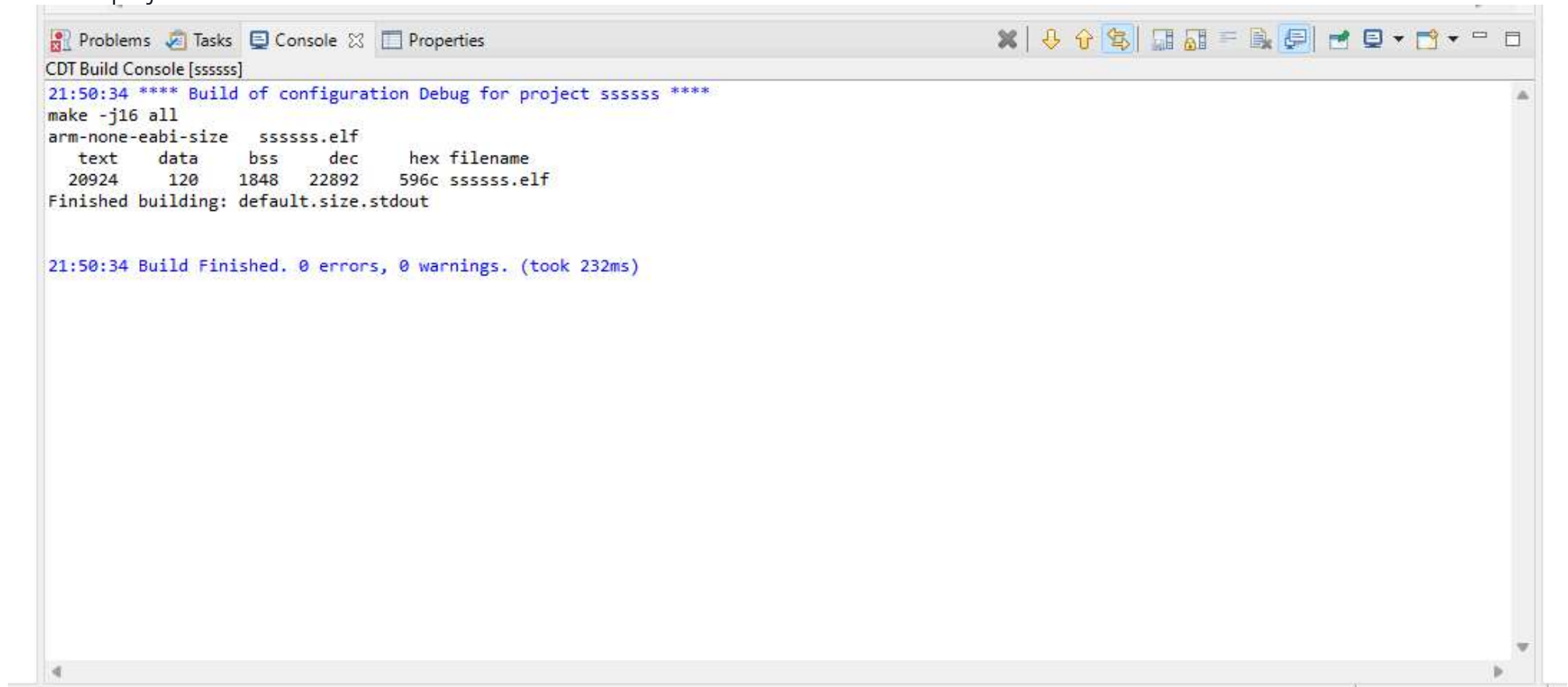
Windows taskbar: 27°, ENG IN, 21:48, 19-03-2023

### 3. use project and build all





## 4. once the project is bulild

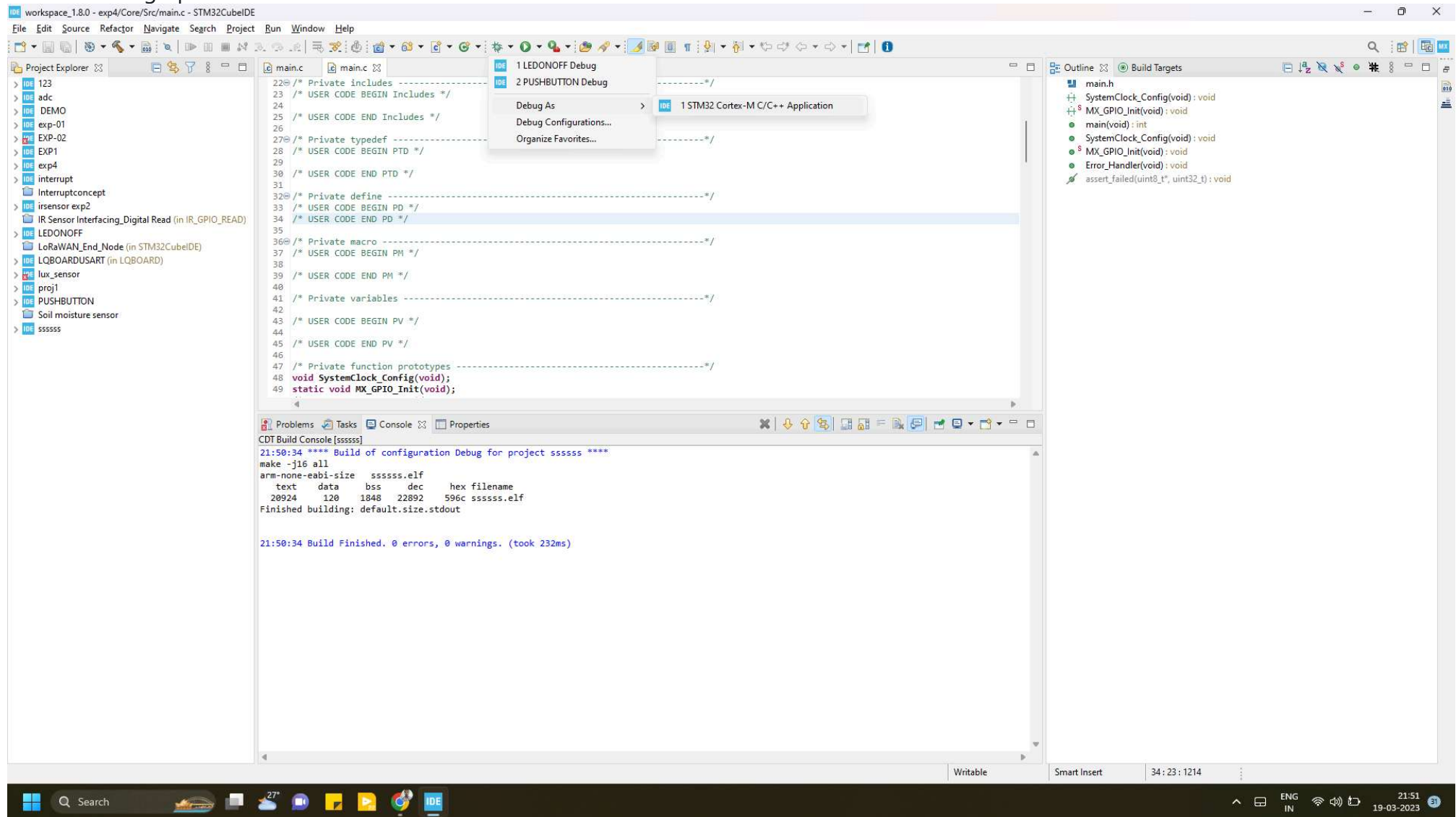


The screenshot shows the CDT Build Console window for a project named 'ssssss'. The console displays the following output:

```
CDT Build Console [ssssss]
21:50:34 **** Build of configuration Debug for project ssssss ****
make -j16 all
arm-none-eabi-size  ssssss.elf
   text    data    bss     dec     hex filename
  20924    120    1848    22892    596c ssssss.elf
Finished building: default.size.stdout

21:50:34 Build Finished. 0 errors, 0 warnings. (took 232ms)
```

## 5. click on debug option



5 connect the ARM board to power supply and usb

6. check for execution of the output using run option

7. Open serial port utility and check the output

## STM 32 CUBE PROGRAM :



```
#if defined (__ICCARM__) || defined (__ARMCC_VERSION)
#define PUTCHAR_PROTOTYPE int fputc(int ch, FILE *f)
#elif defined(__GNUC__)
#define PUTCHAR_PROTOTYPE int __io_putchar(int ch)
#endif

while (1)
{
    printf("your roll no and dept ");
    HAL_Delay(1000);

}

PUTCHAR_PROTOTYPE
{
    HAL_UART_Transmit(&huart2, (uint8_t*)&ch, 1, 0xFFFF);
    return ch;
}
```

**Output screen shots of Serial port utility :**

The screenshot displays the STM32CubeIDE interface. The main editor shows the `main.c` file with the following content:

```

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 */

```

The Project Explorer on the left shows the project structure, including the `src` directory with `main.c` and other files.

The Build Console at the bottom shows the compilation process:

```

arm-none-eabi-size  ex-4.elf
arm-none-eabi-objdump -h -S ex-4.elf > "ex-4.list"
text    data    bss     dec     hex filename
16672   112     2056   18840   4998 ex-4.elf
Finished building: default.size.stdout

Finished building: ex-4.list

09:17:40 Build Finished. 0 errors, 2 warnings. (took 1s.794ms)

```

The Memory Regions panel on the right shows the memory layout for `ex-4.elf`:

Region	Start address	End address	Size
RAM	0x20000000	0x2000ffff	64 KB
RAM2	0x10000000	0x10007fff	32 KB
FLASH	0x08000000	0x0803ffff	256 KB

## Result :

configuring and usart is accomplished and string data is visualized on the serial port utility