ENERGY METER

Modbus RS485 communication

By Aliyan Waseem

System requirement

- Hardware
  - Energy meter (CET PMC-53M-A)
  - STM32f401ret6 (Black pill board)
  - St-link V2
  - RS485 to UART ttl convertor
- Software
  - STM32CubeIDE

Energy meter

- CET PMC-53M-A

fig:1

Why CET?

- Give the all necessary peremetrs of energy.
- ✓ 3-phase voltage line
- ✓ 3-phase current
- ✓ Power factor
- ✓ Energy
- ✓ Harmonics
- Accurate readings
- Easy to use

Microcontroller

- STM32f401ret6 black pill board

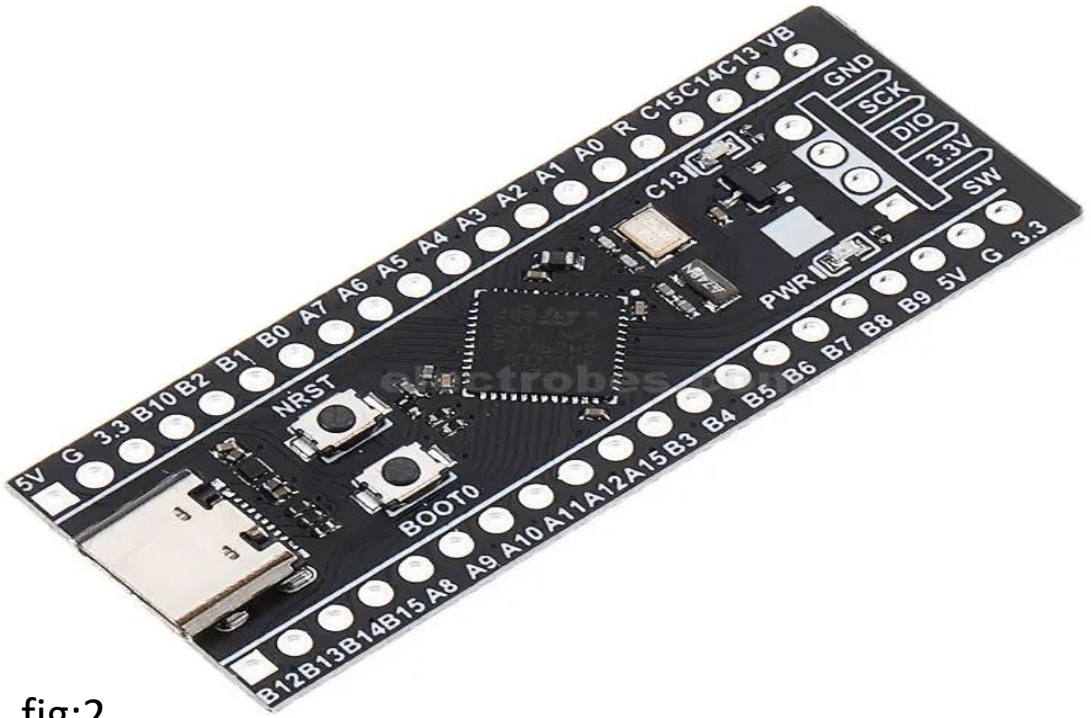

fig:2

Why Black Pill Board

- Low power consumtion
- Decent processing power 84MHz
- Rich number of GPIO
- Moderate Price range

Debugger

St-link V2

fig:3

Why ST-LINK V2?

- Boot loader to upload program to stm board.
- Serial Wire Debugger use to debug code.
- Provide Power to board

Rs485 convertor

Rs485 to uart convertor

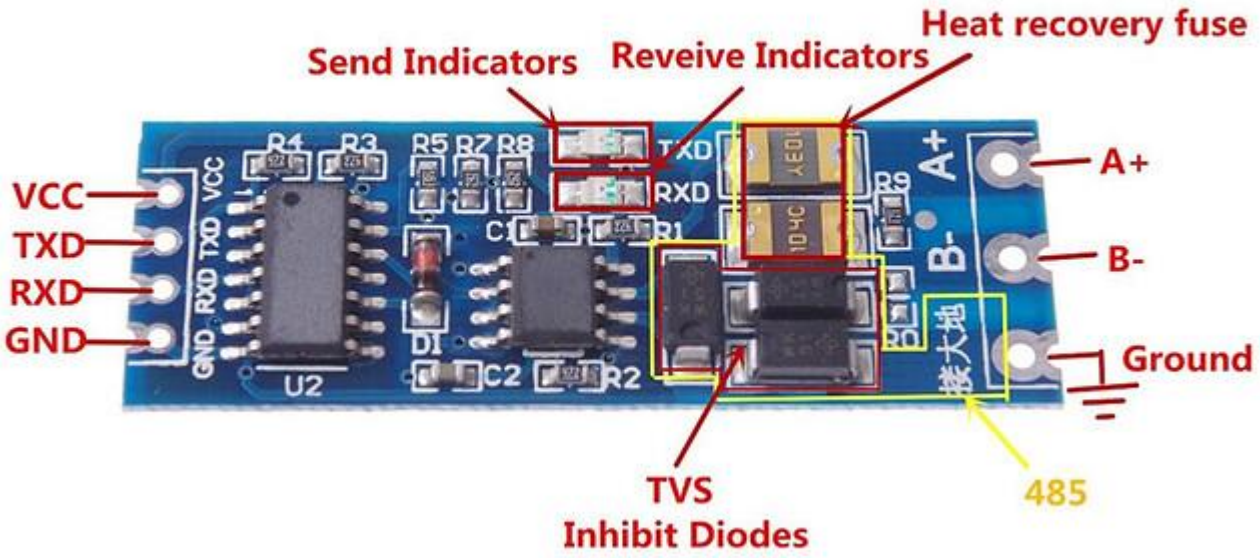

fig:4

Why this?

Modbus rs485 signal can directly malfunctioned the GPIO of stm board so we use rs485 to uart convertor and also it is easy to configure and use.

IDE

STM32CubeIDE

Use to program stm boards

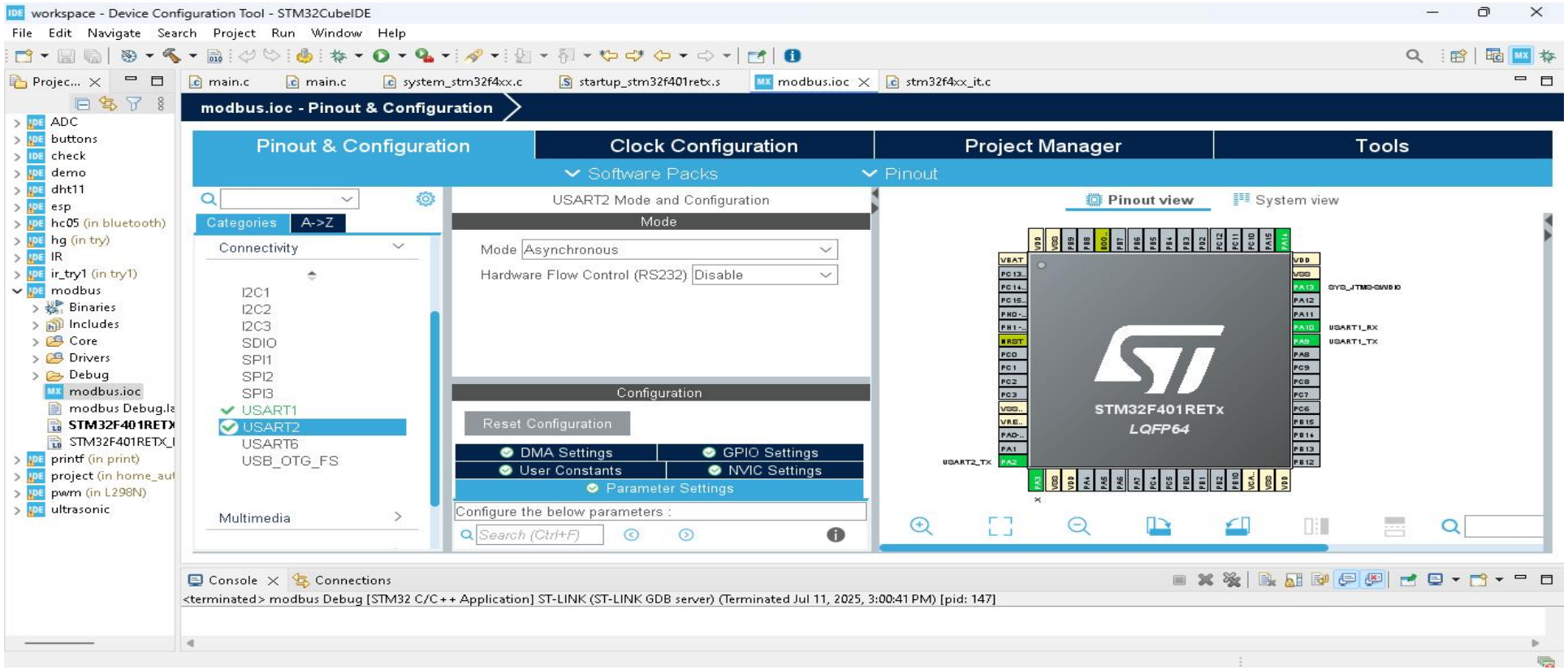

fig:5

Working

- Step -1

Open stm32CubeIDE

In title bar from top left corner click on

file>>new>>stm32project (as shown in fig:6)

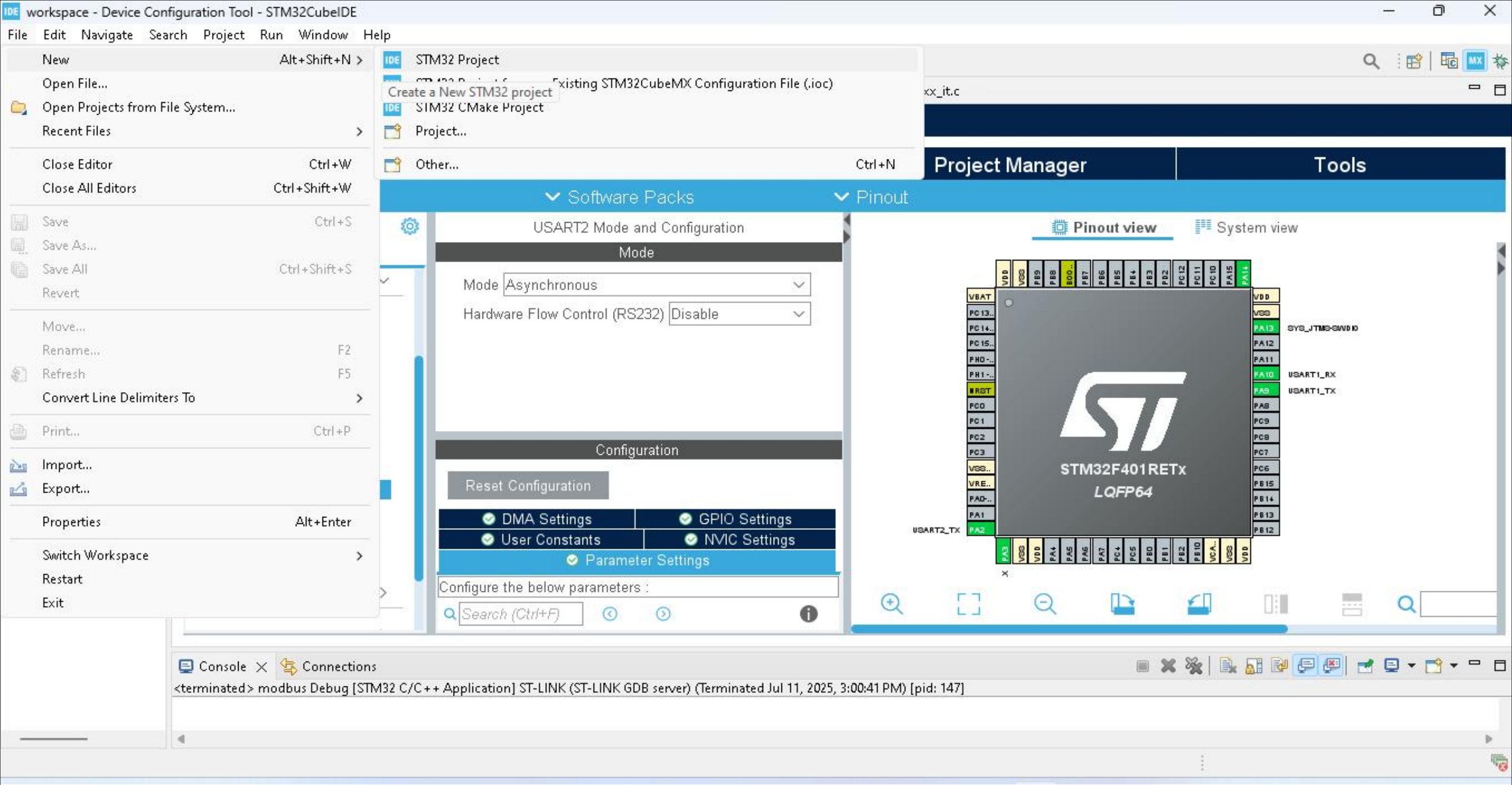
workspace - Device Configuration Tool - STM32CubeIDE
File Edit Navigate Search Project Run Window Help
New Alt+Shift+N
Open File...
Open Projects from File System...
Recent Files
Close Editor Ctrl+W
Close All Editors Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Import...
Export...
Properties Alt+Enter
Switch Workspace
Restart
Exit
STM32 Project
Existing STM32CubeMX Configuration File (.ioc)
Create a New STM32 project
STM32 CMake Project
Project...
Other... Ctrl+N
Project Manager
Tools
Software Packs
Pinout
USART2 Mode and Configuration
Mode
Mode Asynchronous
Hardware Flow Control (RS232) Disable
Configuration
Reset Configuration
DMA Settings
GPIO Settings
User Constants
NVIC Settings
Parameter Settings
Configure the below parameters :
Search (Ctrl+F)
Pinout view
System view
STM32F401RETx
LQFP64
USART2_TX
USART1_RX
USART1_TX
Console
Connections
<terminated> modbus Debug [STM32 C/C++ Application] ST-LINK (ST-LINK GDB server) (Terminated Jul 11, 2025, 3:00:41 PM) [pid: 147]

fig:6

- Step-2

A new pop up window appear

In commercial port number window enter board number in our case it is stm32f401ret6.

Click the board from MPU list at right side and click next

(as shown in fig:7)

STM32 Project

Target Selection

STM32 target or STM32Cube example selection is required

IDE

MCU/MPU Selector | Board Selector | Example Selector | Cross Selector

MCU/MPU Filters

Commercial Part Number

PRODUCT INFO

- Segment
- Series
- Line
- Marketing Status
- Price
- Package
- Core
- Coprocessor

Features | Block Diagram | Docs & Resources | CAD Resources | Datasheet | Buy

MCUs/MPUs List: 4801 items

Export

*	Comme...	Part No	Reference	Marketin...	Unit Price...	Board	Package	Flash	RAM	I/O	Frequency
☆	STM32C0...	STM32C0...	STM32C0...	Coming so...	NA		WLCSP 1...	32 kBytes	6 kBytes	18	48 MHz
☆	STM32C0...		STM32C0...	Active	0.2812		WLCSP 1...	32 kBytes	6 kBytes	18	48 MHz
☆	STM32C0...		STM32C0...	Coming so...	NA		TSSOP-20	16 kBytes	6 kBytes	18	48 MHz
☆	STM32C0...		STM32C0...	Active	0.2727		TSSOP-20	16 kBytes	6 kBytes	18	48 MHz
☆	STM32C0...		STM32C0...	Active	0.2918		TSSOP-20	16 kBytes	6 kBytes	18	48 MHz
☆	STM32C0	STM32C0	STM32C0	Active	0.2918		TSSOP-20	16 kBytes	6 kBytes	18	48 MHz

< Back | Next > | Finish | Cancel

fig:7

- Step-3

A new pop up window appear type your project name and hit finish.

- Step-4

A new window appear click on system core drop down in left side bar (as shown in fig:8)

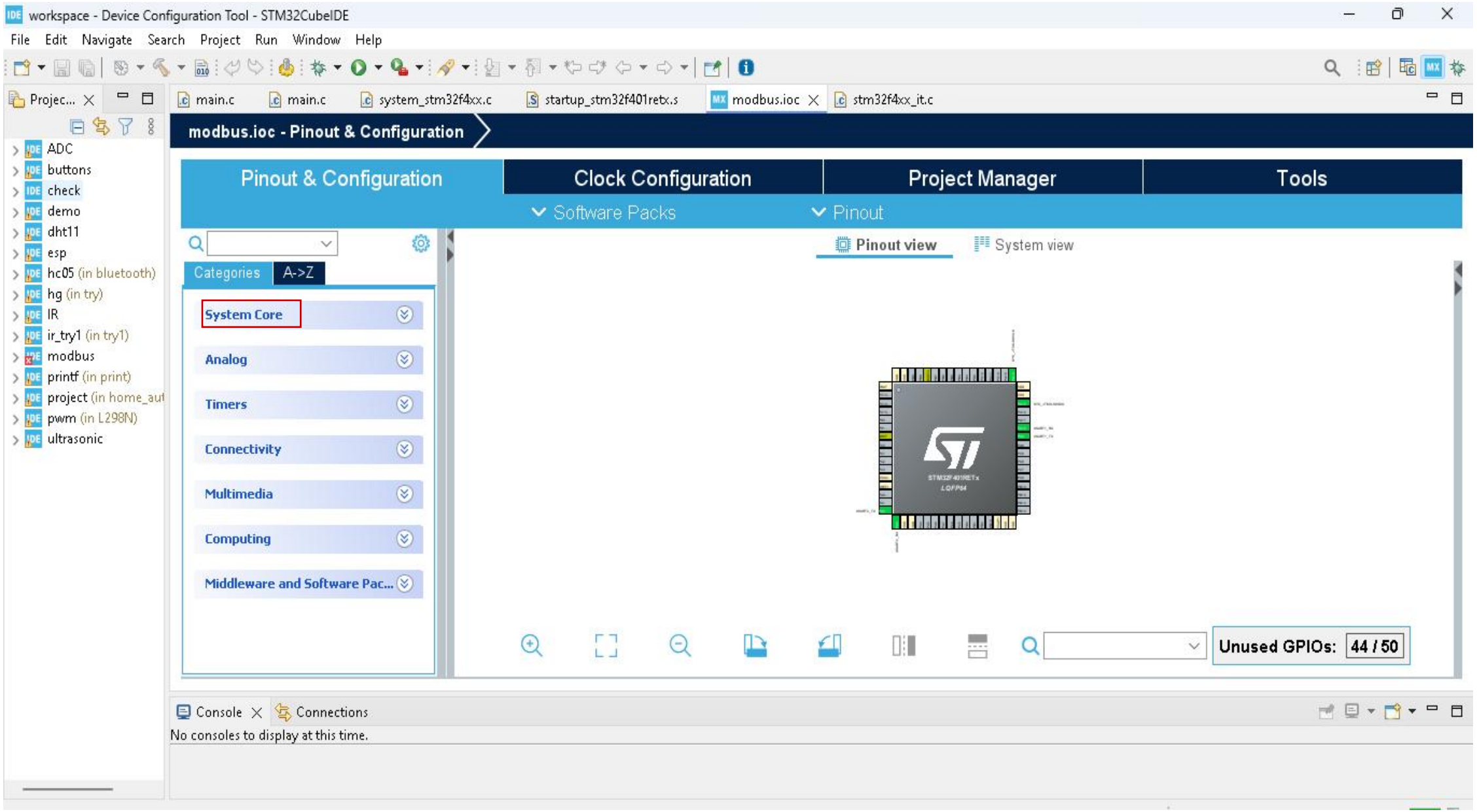
workspace - Device Configuration Tool - STM32CubeIDE
File Edit Navigate Search Project Run Window Help
main.c
main.c
system_stm32f4xx.c
startup_stm32f401retx.s
modbus.ioc
stm32f4xx_it.c
modbus.ioc - Pinout & Configuration
Pinout & Configuration
Clock Configuration
Project Manager
Tools
Software Packs
Pinout
Pinout view
System view
Categories
A->Z
System Core
Analog
Timers
Connectivity
Multimedia
Computing
Middleware and Software Pac...
Unused GPIOs: 44 / 50
Console
Connections
No consoles to display at this time.
ADC
buttons
check
demo
dht11
esp
hc05 (in bluetooth)
hg (in try)
IR
ir_try1 (in try1)
modbus
printf (in print)
project (in home_aut
pwm (in L298N)
ultrasonic

fig:8

- Step-5

From drop down click on sys

(as shown in fig:9)

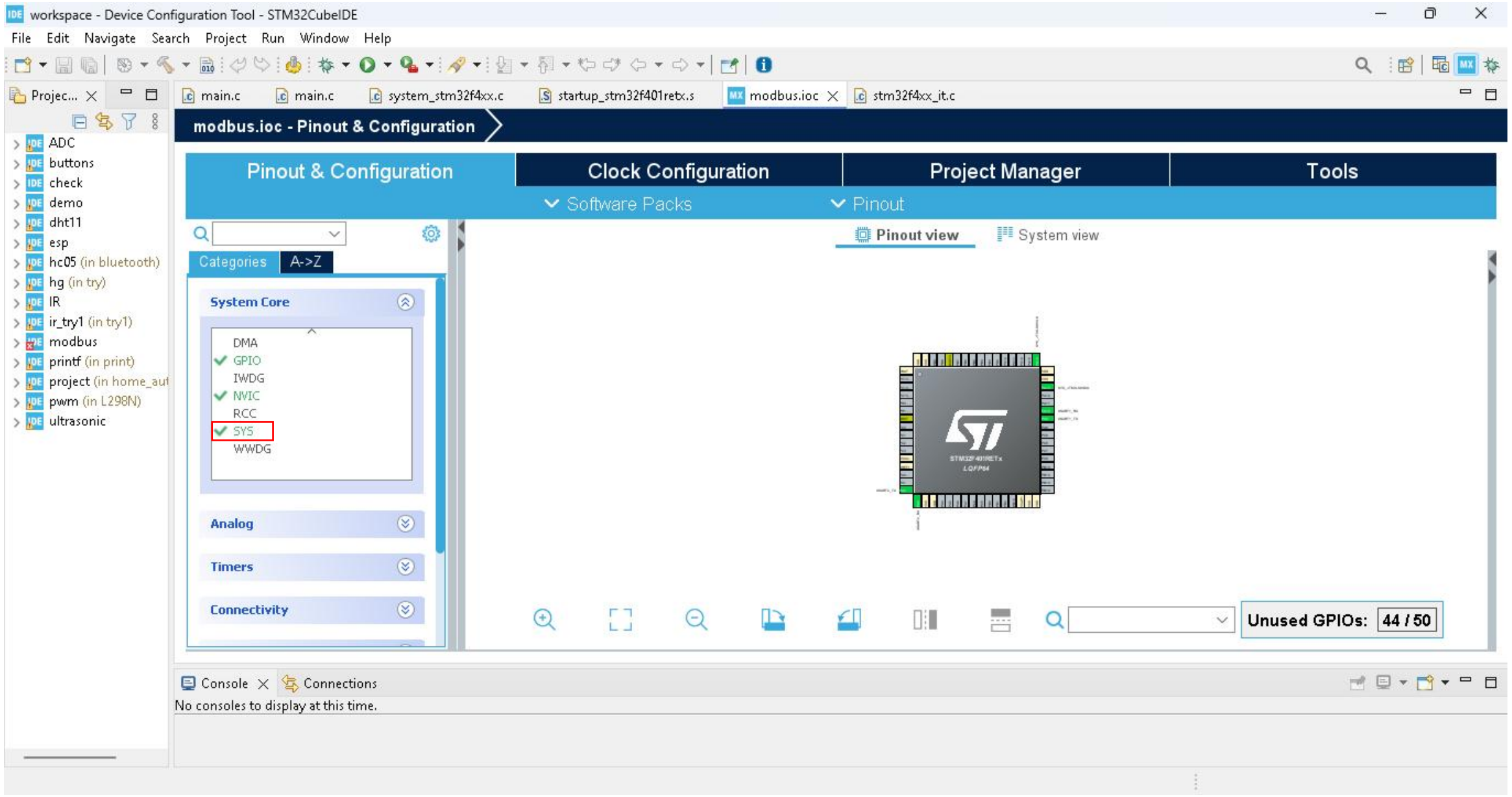
workspace - Device Configuration Tool - STM32CubeIDE
File Edit Navigate Search Project Run Window Help
main.c
main.c
system_stm32f4xx.c
startup_stm32f401retx.s
modbus.ioc
stm32f4xx_it.c
modbus.ioc - Pinout & Configuration
Pinout & Configuration
Clock Configuration
Project Manager
Tools
Software Packs
Pinout
Pinout view
System view
Categories
A->Z
System Core
DMA
GPIO
IWDG
NVIC
RCC
SYS
WWDG
Analog
Timers
Connectivity
Unused GPIOs: 44 / 50
Console
Connections
No consoles to display at this time.
ADC
buttons
check
demo
dht11
esp
hc05 (in bluetooth)
hg (in try)
IR
ir_try1 (in try1)
modbus
printf (in print)
project (in home_aut
pwm (in L298N)
ultrasonic

fig:9

- Step-6

After clicking on sys a new side bar appear in right side of sys from here click on serial wire from debug section. (as shown in fig:10)

This will enable serial wire pins of stm board used for program loading and debugging with st link v2.

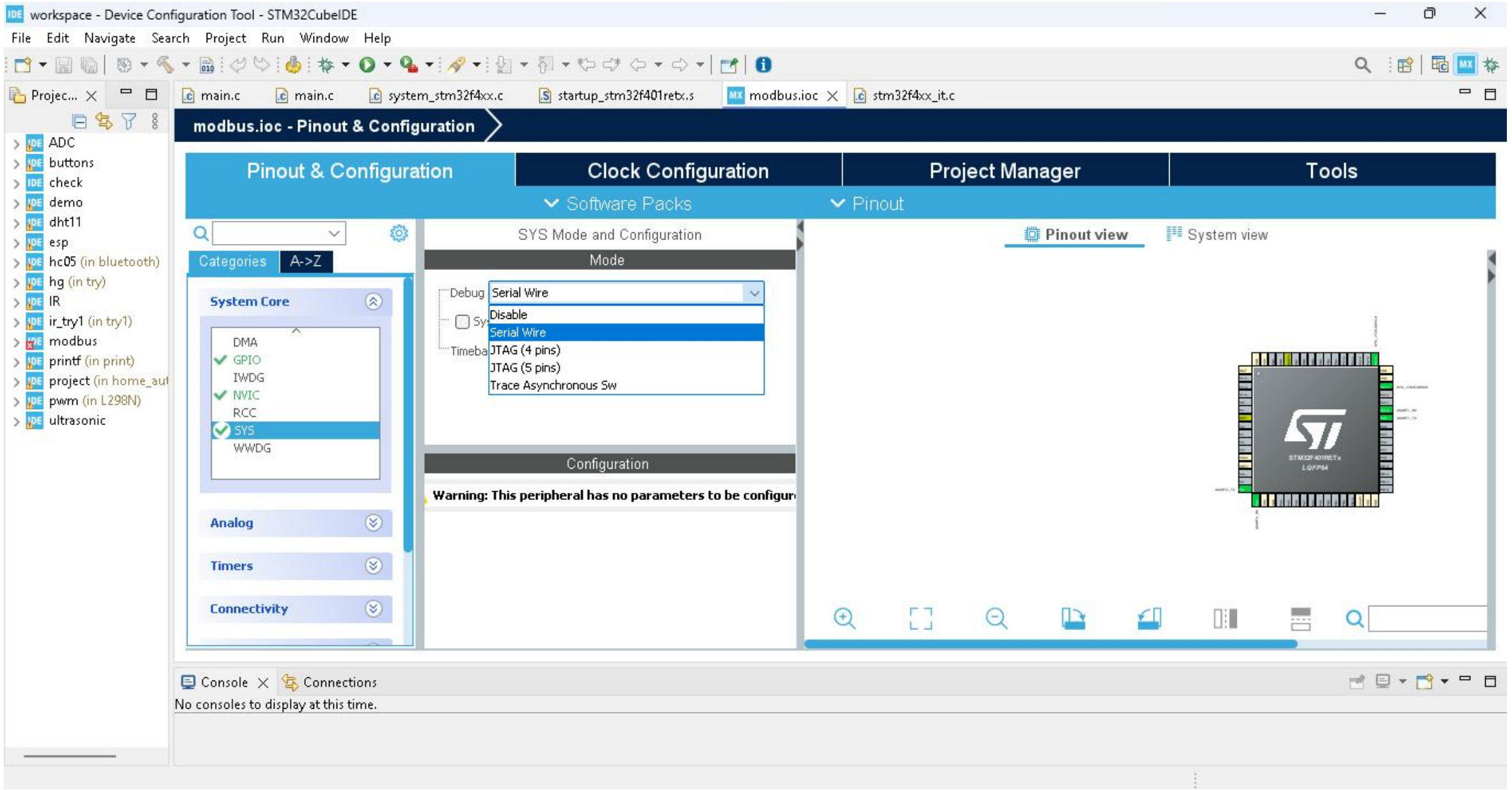
workspace - Device Configuration Tool - STM32CubeIDE
File Edit Navigate Search Project Run Window Help
main.c
main.c
system_stm32f4xx.c
startup_stm32f401retx.s
modbus.ioc
stm32f4xx_it.c
modbus.ioc - Pinout & Configuration
Pinout & Configuration
Clock Configuration
Project Manager
Tools
Software Packs
Pinout
SYS Mode and Configuration
Mode
Debug
Serial Wire
Disable
Serial Wire
JTAG (4 pins)
JTAG (5 pins)
Trace Asynchronous Sw
Configuration
Warning: This peripheral has no parameters to be configur
Categories
A->Z
System Core
DMA
GPIO
IWDG
NVIC
RCC
SYS
WWDG
Analog
Timers
Connectivity
Pinout view
System view
Console
Connections
No consoles to display at this time.
ADC
buttons
check
demo
dht11
esp
hc05 (in bluetooth)
hg (in try)
IR
ir_try1 (in try1)
modbus
printf (in print)
project (in home_aut
pwm (in L298N)
ultrasonic

fig:10

- Step-7

Now we have to enable uart gpio from left side bar click on connectivity >> UART1

A new side bar window appear on very right side of this bar

Here from mode select ‘Asynchronous’ (as shown fig:11)

Now from configuration select parameter setting

from there set baud rate ‘9600’

set word length 9 bits (for 8 bit data and 1 bit for parity)

set parity even (as CET energy meter uses even parity for modbus)

(as shown in fig:11)

and do the same settings for UART2 to send data to esp

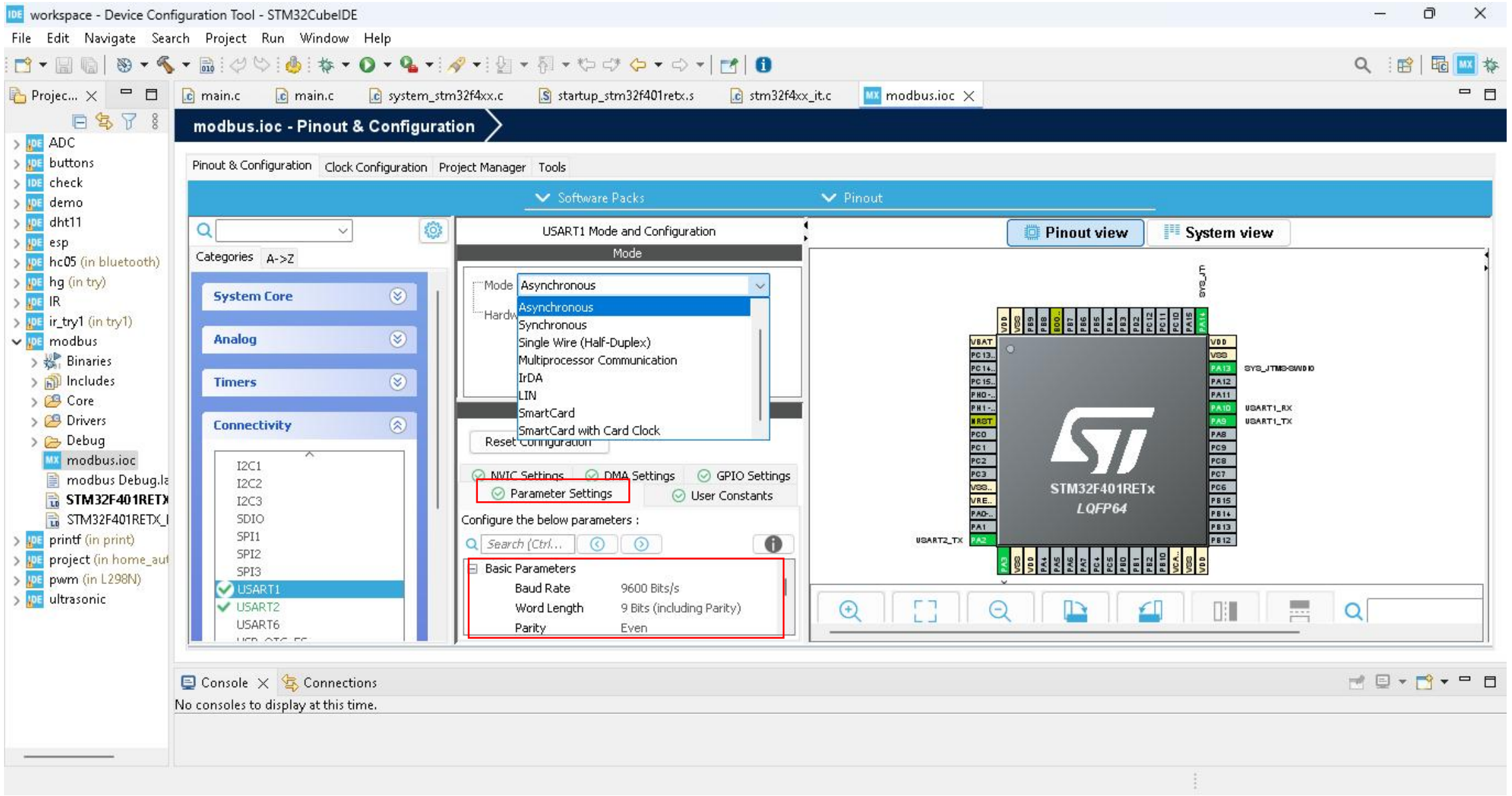
workspace - Device Configuration Tool - STM32CubeIDE
File Edit Navigate Search Project Run Window Help
main.c
main.c
system_stm32f4xx.c
startup_stm32f401retx.s
stm32f4xx_it.c
modbus.ioc
modbus.ioc - Pinout & Configuration
Pinout & Configuration
Clock Configuration
Project Manager
Tools
Software Packs
Pinout
ADC
buttons
check
demo
dht11
esp
hc05 (in bluetooth)
hg (in try)
IR
ir_try1 (in try1)
modbus
Binaries
Includes
Core
Drivers
Debug
modbus.ioc
printf (in print)
project (in home_aut
pwm (in L298N)
ultrasonic
Categories
A->Z
System Core
Analog
Timers
Connectivity
I2C1
I2C2
I2C3
SDIO
SPI1
SPI2
SPI3
USART1
USART2
USART6
USART1 Mode and Configuration
Mode
Mode
Asynchronous
Asynchronous
Synchronous
Single Wire (Half-Duplex)
Multiprocessor Communication
IrDA
LIN
SmartCard
SmartCard with Card Clock
NVIC Settings
DMA Settings
GPIO Settings
Parameter Settings
User Constants
Configure the below parameters :
Search (Ctrl...
Basic Parameters
Baud Rate 9600 Bits/s
Word Length 9 Bits (including Parity)
Parity Even
Pinout view
System view
STM32F401RETx
LQFP64
USART1_RX
USART1_TX
USART2_TX
Console
Connections
No consoles to display at this time.

fig:11

- Step-8

Now hit save(Ctrl+S)

and a new pop up appear which ask to generate code click ok (yes)

this will generate the configuration code for our project and then “open the prospective now” option appear click ok and main.c file is open.

- Step-9

Code

now add this code in respective commented part

```
/* USER CODE BEGIN Includes */
#include <stdio.h>
#include <stdbool.h>
#include <string.h>
#include <stdlib.h>
#include <stdint.h>
#include <math.h>
/* USER CODE END Includes */
```

```
/* USER CODE BEGIN PD */
#define MODBUS_MAX_BUFFER 256
/* USER CODE END PD */

/* USER CODE BEGIN PV */
// for debugging
int debug=0;
int debug1=0;
int debug2=0;

// for modbus communication
uint8_t modbusRxBuffer[MODBUS_MAX_BUFFER]={0};
uint8_t modbusTxBuffer[MODBUS_MAX_BUFFER]={0};
uint16_t modbusRxIndex = 0;
```

```
bool modbusFrameReceived = false;

uint8_t recieved_data[MODBUS_MAX_BUFFER] = {0};
uint16_t crcCalculated=0;
uint16_t crcReceived=0;

// for voltage reading
float prevVolt = 0.0;
float currentVolt = 0.0;
float check = 0.0;

uint32_t ieee = 0;
uint32_t ieee2 = 0;
uint8_t floatBytes[4]={0};
```

```
int byteCount = 0;
int numFloats  =0;
float values[10]={0};
int baseIndex = 0;
uint32_t ieeeVal=0;

/* USER CODE END PV */

/* USER CODE BEGIN PFP */
// to copy data frame in a buffer before receiving the next data frame
void CopyData();
/* USER CODE END PFP */

/* Private user code ---------------------------------------------------*/
/* USER CODE BEGIN 0 */
// to calculate the 16-bit crc
```

```
uint16_t Modbus_CRC16(uint8_t *buffer, uint16_t length){
   uint16_t crc = 0xFFFF; // initial value of crc always 0xffff
   for (uint16_t i = 0; i < length; i++){
      crc ^= buffer[i]; // take xor of every byte with previous crc
      for (uint8_t j = 0; j < 8; j++)  // shift CRC right 8 time
      {
         if (crc & 0x0001) // if lsb is 1
            crc = (crc >> 1) ^ 0xA001; // first right shift and then take xor with 0xa001
         else // if lsb is 0
            crc = crc >> 1; // right shift the crc// repeat this task 8 times
      }// repeat this task for every byte of data frame
   }
return crc;
}
```

```
void Modbus_SendFrame(uint8_t slaveID, uint8_t functionCode, uint16_t startAddress, uint16_t quantity)
{
    uint8_t index = 0; // number of bytes in data frame that is to be send
    modbusTxBuffer[index++] = slaveID;
    modbusTxBuffer[index++] = functionCode;
    modbusTxBuffer[index++] = (startAddress >> 8) & 0xFF; //shift for high byte from 2 bytes and take & for only 1 byte
    modbusTxBuffer[index++] = startAddress & 0xFF; // take & with 1 byte to take only 1 byte from 2 bytes
    modbusTxBuffer[index++] = (quantity >> 8) & 0xFF;
    modbusTxBuffer[index++] = quantity & 0xFF;
    uint16_t crc = Modbus_CRC16(modbusTxBuffer, index);
    modbusTxBuffer[index++] = crc & 0xFF;        // CRC Low
    modbusTxBuffer[index++] = (crc >> 8) & 0xFF;  // CRC High
    HAL_UART_Transmit(&huart1, modbusTxBuffer, index, HAL_MAX_DELAY);
}
```

```
// Function to convert IEEE 754 32-bit float representation to decimal
float ieee754_to_float(uint32_t ieee){
            // as it is value = (-1)^s * F * 2^E
   int sign = (ieee >> 31) & 0x1;
   int exponent = ((ieee >> 23) & 0xFF) - 127;  // Remove bias  (E = exponent - bias for single precision bias is 127)
   uint32_t mantissa = ieee & 0x7FFFFF;
   float value = 1.0; // implied leading 1 for normalized numbers
   for (int i = 0; i < 23; i++){
      if (mantissa & (1 << (22 - i))){
         value += pow(2, -(i + 1)); // to calculate the fraction part (1+Fraction) ; fraction = decimal of mantissa in points}   }
   float result = value * pow(2, exponent);
   if (sign) result = -result;
   return result;
}
/* USER CODE END 0 */
```

```
/* USER CODE BEGIN 2 */
  uint8_t slaveID = 0x01;
  uint8_t functionCode = 0x03;
  uint16_t startAddress = 0x0000;
  uint16_t quantity = 0x0012;
  HAL_UARTEx_ReceiveToIdle_IT(&huart1, modbusRxBuffer, MODBUS_MAX_BUFFER);
  Modbus_SendFrame(slaveID, functionCode, startAddress, quantity);
  /* USER CODE END 2 */
```

Now in while loop add this

```
if (modbusFrameReceived)
                {
                  modbusFrameReceived = false;
```

```
                debug=12;
                if (modbusRxIndex >= 5 && modbusRxIndex <= MODBUS_MAX_BUFFER)
                {
                        debug2=51;
                  crcReceived = (recieved_data[modbusRxIndex - 1] << 8)
|(recieved_data[modbusRxIndex - 2]);
                  crcCalculated = Modbus_CRC16(recieved_data, modbusRxIndex - 2);
                  if (crcReceived == crcCalculated)
                  {
                        // assuming only reading floating type data of only 1 voltage line
                        byteCount = recieved_data[2];  // Number of data bytes
                        numFloats = byteCount / 4;
                        for (int i = 0; i < numFloats; i++)
                        {
                           baseIndex = 3 + i * 4;
```

```
  ieeeVal =
                              ((uint32_t)recieved_data[baseIndex]     << 24) |
                              ((uint32_t)recieved_data[baseIndex + 1] << 16) |
                              ((uint32_t)recieved_data[baseIndex + 2] << 8)  |
                              ((uint32_t)recieved_data[baseIndex + 3]);
                            values[i] = ieee754_to_float(ieeeVal);
}

else
                    {
                        // Invalid CRC
                    }
                  }
              }
    /* USER CODE END WHILE */
```

```
/* USER CODE BEGIN 4 */

void CopyData(){
        for(int i=0;i<modbusRxIndex;i++){
                recieved_data[i]=modbusRxBuffer[i];}  }

void HAL_UARTEx_RxEventCallback(UART_HandleTypeDef *huart, uint16_t Size){
        modbusRxIndex=Size;
        CopyData();
        modbusFrameReceived=true;
        memset(modbusRxBuffer, 0, MODBUS_MAX_BUFFER);
        HAL_UARTEx_ReceiveToIdle_IT(&huart1, modbusRxBuffer, MODBUS_MAX_BUFFER);
}
/* USER CODE END 4 */
```

- Step-10

Now we are done with code now next is to save it press(ctrl+s).

and do the physical connection of wires. (as shown in fig:12)

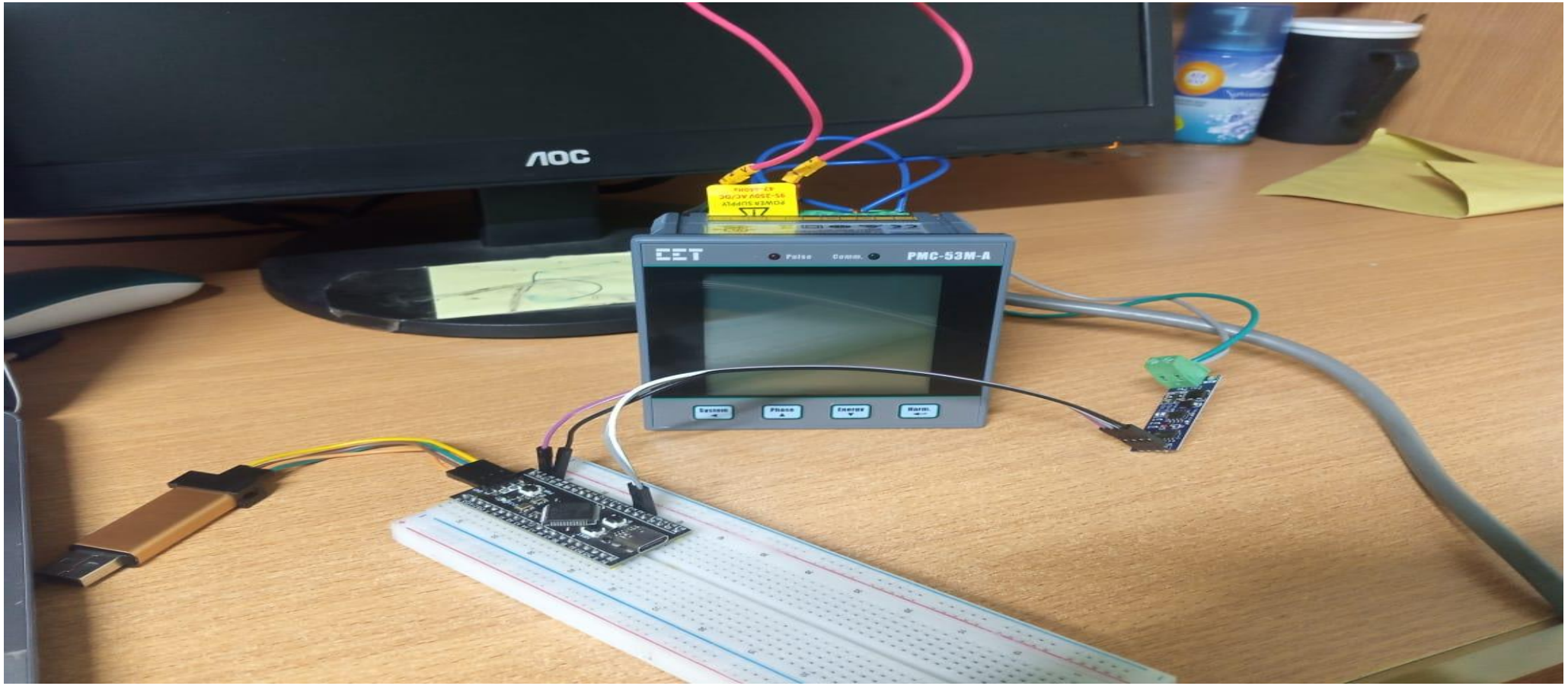

fig:12

- Step-11

Connect ST-link to usb port of laptop

press the debug button from tittle bar a new window appear just click next and finish.

(as shown in fig:13)

The program first compiled and then after successfully compilation a window open and then a pop up appear to switch the perspective

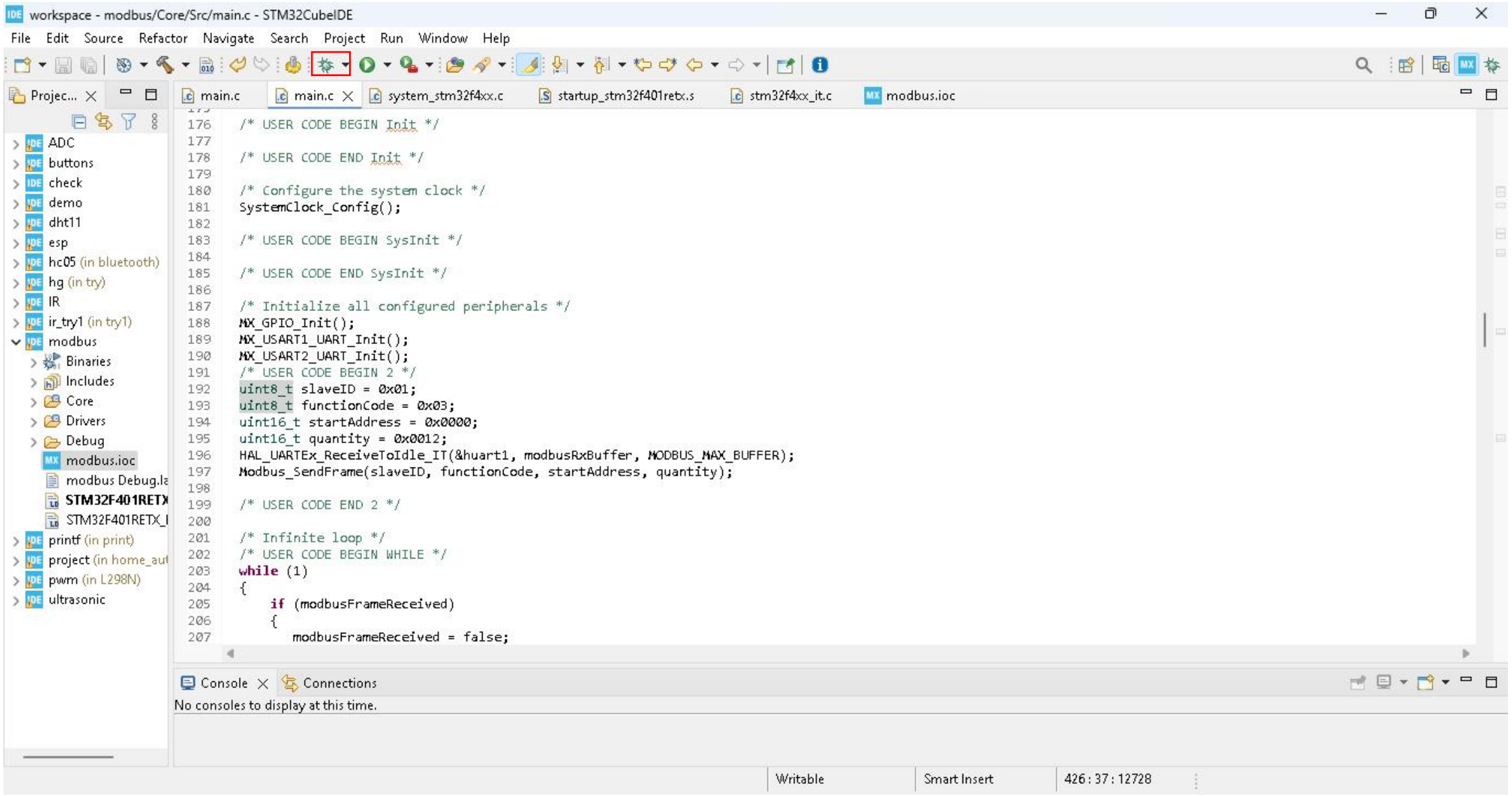
workspace - modbus/Core/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Projec...
ADC
buttons
check
demo
dht11
esp
hc05 (in bluetooth)
hg (in try)
IR
ir_try1 (in try1)
modbus
Binaries
Includes
Core
Drivers
Debug
modbus.ioc
modbus Debug.la
STM32F401RETX
STM32F401RETX_I
printf (in print)
project (in home_aut
pwm (in L298N)
ultrasonic
main.c
main.c
system_stm32f4xx.c
startup_stm32f401retx.s
stm32f4xx_it.c
modbus.ioc
176 /* USER CODE BEGIN Init */
177
178 /* USER CODE END Init */
179
180 /* Configure the system clock */
181 SystemClock_Config();
182
183 /* USER CODE BEGIN SysInit */
184
185 /* USER CODE END SysInit */
186
187 /* Initialize all configured peripherals */
188 MX_GPIO_Init();
189 MX_USART1_UART_Init();
190 MX_USART2_UART_Init();
191 /* USER CODE BEGIN 2 */
192 uint8_t slaveID = 0x01;
193 uint8_t functionCode = 0x03;
194 uint16_t startAddress = 0x0000;
195 uint16_t quantity = 0x0012;
196 HAL_UARTEx_ReceiveToIdle_IT(&huart1, modbusRxBuffer, MODBUS_MAX_BUFFER);
197 Modbus_SendFrame(slaveID, functionCode, startAddress, quantity);
198
199 /* USER CODE END 2 */
200
201 /* Infinite loop */
202 /* USER CODE BEGIN WHILE */
203 while (1)
204 {
205 if (modbusFrameReceived)
206 {
207 modbusFrameReceived = false;
Console
Connections
No consoles to display at this time.
Writable
Smart Insert
426 : 37 : 12728

fig:13

- Step-12

In the right side bar in live expression option type the name of variables and arrays in it and press the resume button from tittle bar.

(as shown in fig:14)

You will get the values like that

Expression	Type	Value
modbusTxBuffer	uint8_t [256]	[256]
recieved_data	uint8_t [256]	[256]
debug	int	12
debug1	int	43
debug2	int	51
crcReceived	uint16_t	10389
crcCalculated	uint16_t	10389
modbusRxIndex	uint16_t	9
modbusFrameRe	_Bool	false
ieee	uint32_t	1130931514
currentVolt	float	232.645416
floatBytes	uint8_t [4]	[4]
prevVolt	float	0
check	float	232.645416
ieee2	uint32_t	1130931514
Add new express		

values	float [10]	[10]
values[0]	float	231.010651
values[1]	float	230.802322
values[2]	float	135.302628
values[3]	float	199.038528
values[4]	float	5.87747175e-039
values[5]	float	97.0421906
values[6]	float	97.2550049
values[7]	float	64.7657318
values[8]	float	5.87747175e-039
values[9]	float	0

fig:14