Sensors used in Driver Alert System Using CAN

1. **LM35 Tempreture Sensor**

LM35 temperature sensor in a driver alert system or similar applications because it is a simple, reliable, and low-cost way to measure temperature — particularly for detecting changes in body temperature or cabin temperature that may indicate driver fatigue, discomfort, or unsafe conditions.

1. **To Monitor Driver's Body Temperature**
   - **Why?** Slight drops in skin temperature can indicate **fatigue or drowsiness**.
   - **Example:** If the LM35 is placed on the steering wheel or seat, it can measure the driver's skin contact temperature.
   - **Alert trigger:** Unusual temperature patterns could suggest a drop in alertness.
2. **To Monitor Cabin Temperature**
   - **Why?** An **overheated or very cold cabin** can cause discomfort and reduce driver focus.
   - **Example:** If the LM35 detects a cabin temperature above 35°C, the system can suggest turning on the AC or opening a window.
   - **Safety feature:** Ensures optimal thermal comfort, which supports better driving performance.
3. **Ease of Use and Integration**
   - **Analog Output**: LM35 provides a linear analog voltage (10 mV per °C), making it easy to read using an ADC (e.g., Arduino or microcontroller).
   - **No Calibration Required**: It's factory calibrated and provides accurate readings without needing extra setup.
   - **Low Power Consumption**: Suitable for embedded or battery-powered systems.

Feature	Value
Temperature Range	Range−55°C to
Accuracy	±0.5°C (typical) at room temp
Output Voltage	10 mV per °C
Supply Voltage	4V to 30V
Power Consumption	Very Low

4. Pin Out in STM32F407

Function	STM32 Pin (Example: STM32F103C8T6)
LM35 Output	PA0 (ADC1_IN0)
CAN TX	PB9
CAN RX	PB8

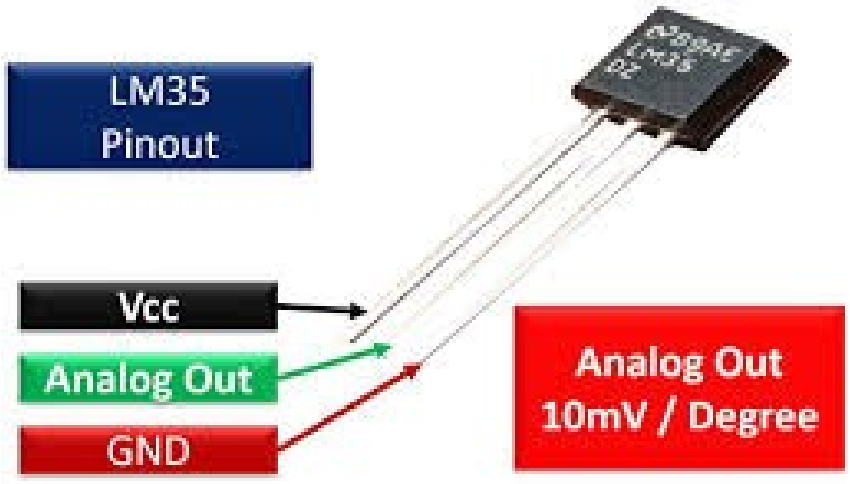


Fig1.1-LM35 Tempreture Sensor

STM32 LM35 Temperature Sensor Interfacing

The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly proportional to the Centigrade temperature. It has a linear

+ 10-mV/°C Scale Factor. Which means you'll have to measure its output voltage and divide it by 0.01 to get the temperature reading in °C, that's all.
The LM35 temperature sensor can be used in a couple of configurations. The basic one is the full positive temperature range (from 2°C up to +150°C). And the full range that can go below zero degrees (from -55°C up to +150°C). The basic configuration requires no external components besides the LM35 itself, and the full range configuration requires an additional resistor which can be calculated using the formula in the diagram below.

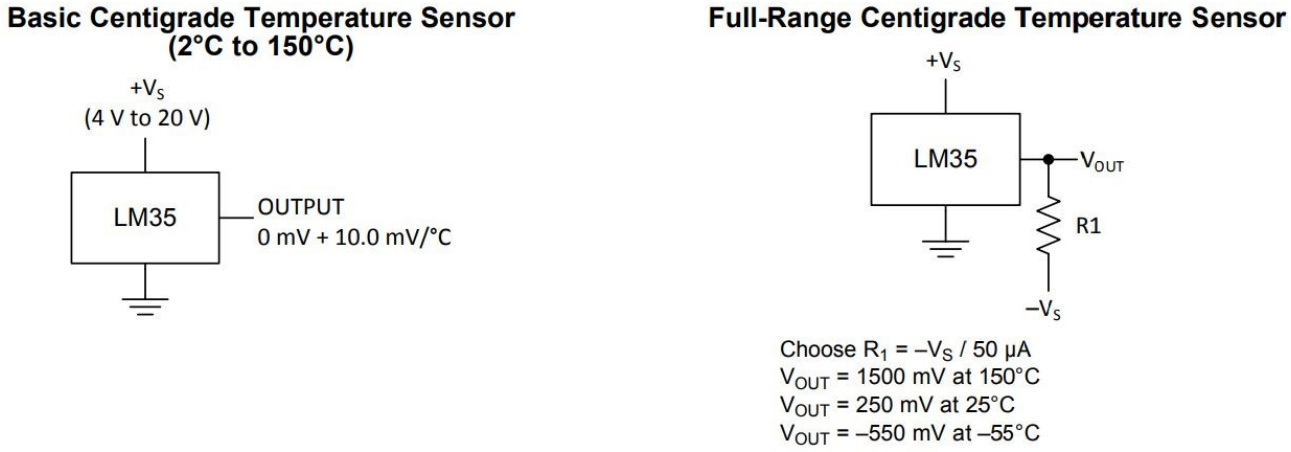


Fig 1.2

2)HCSR04 Ultrasonic Sensor:

The HC-SR04 Ultrasonic Sensor is a widely used distance-measuring sensor that works by sending and receiving ultrasonic sound waves. It is often used in robotics, parking systems, driver alert systems, and object detection.

Pin	Name	Function
1	VCC	Power Supply (+5V)
2	Trig	Trigger Input (Start measurement)
3	Echo	Echo Output (Pulse duration = distance)
4	GND	Ground

HC-SR04 Sensor Features

- •Operating voltage: +5V
- •Theoretical Measuring Distance: 2cm to 450cm
- •Practical Measuring Distance: 2cm to 80cm
- •Accuracy: 3mm
- •Measuring angle covered: <15°
- •Operating Current: <15mA
- •Operating Frequency: 40Hz

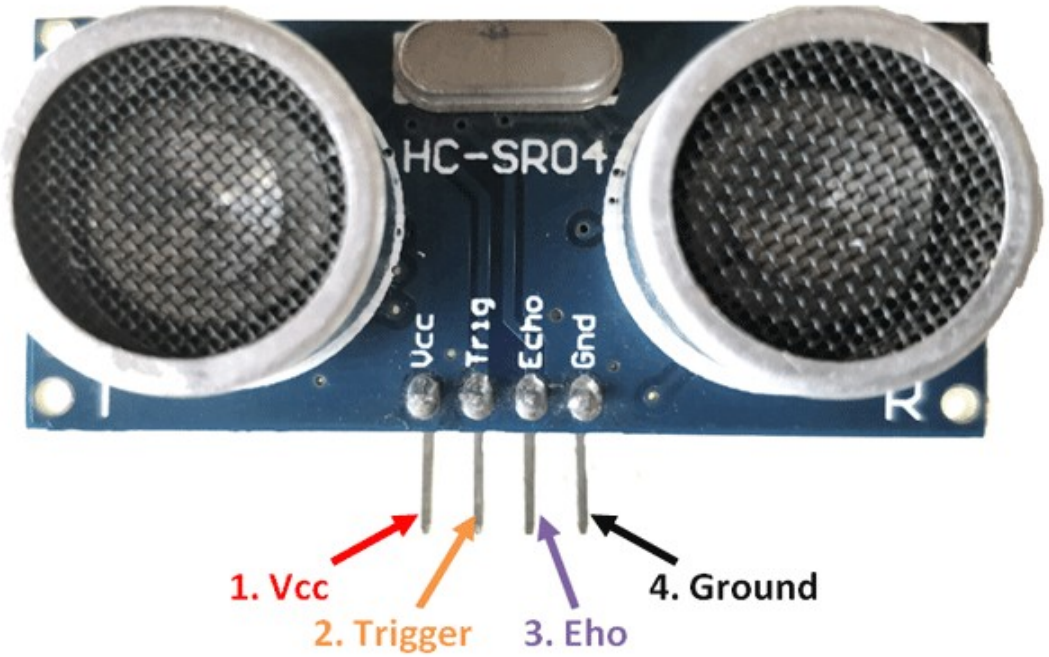


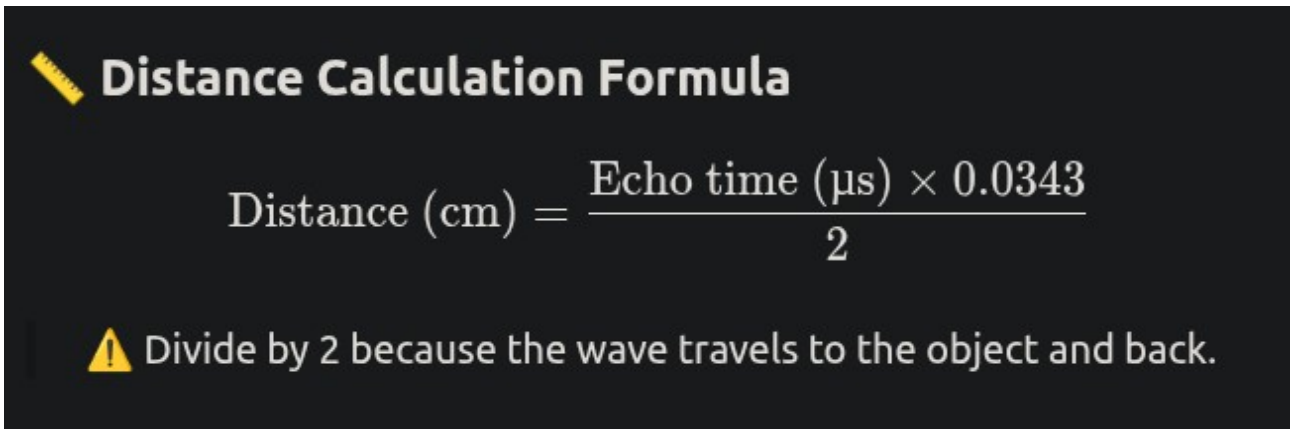
Fig 2.1 HCSR04 Ultrasonic Sensor

Working Principle:

Send a trigger pulse to the Trig pin (at least 10 µs).

- The sensor emits an **8-cycle ultrasonic burst** at 40 kHz.
- Sound reflects off an object and returns to the sensor.
- The sensor sets the Echo pin **HIGH** for the duration of the echo's round-trip time.

- **Microcontroller measures this HIGH time** to calculate the distance.



Purpose	Use Case
Driver proximity alert	Detects nearby obstacles or tailgating
Reverse parking system	Alerts driver when close to wall or car
In-cabin movement sensing	Detect child or pet left in parked car
Blind spot assistance	Mounted on side/rear for object detection

HC-SR04 Ultrasonic Sensor - Working

As shown above the **HC-SR04 Ultrasonic (US) sensor** is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

Distance = Speed × Time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor

this reflected wave is observed by the Ultrasonic receiver module as shown in the picture below

Fig 2.2 – HC-SR04 Ultrasonic Sensor Receiver

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor.

3)<u>I2C serial interface 20 x 4 LCD module :</u>

This is I2C interface 20x4 LCD display module, a new high-quality 4 line 20 character LCD module with on-board contrast control adjustment, backlight and I2C communication interface. For Arduino beginners, no more cumbersome and complex LCD driver circuit connection. The real significance advantages of this I2C Serial LCD module will simplify the circuit connection.

Features :

Compatible with micro-controller board with I2C bus.

Display Type: Black on yellow green backlight.

I2C Address:0x38-0x3F (0x3F default)

Supply voltage: 5V

Interface: I2C to 4bits LCD data and control lines.

Contrast Adjustment : built-in Potentiometer.

Backlight Control: Firmware or jumper wire.

Board Size: 98x60 mm.

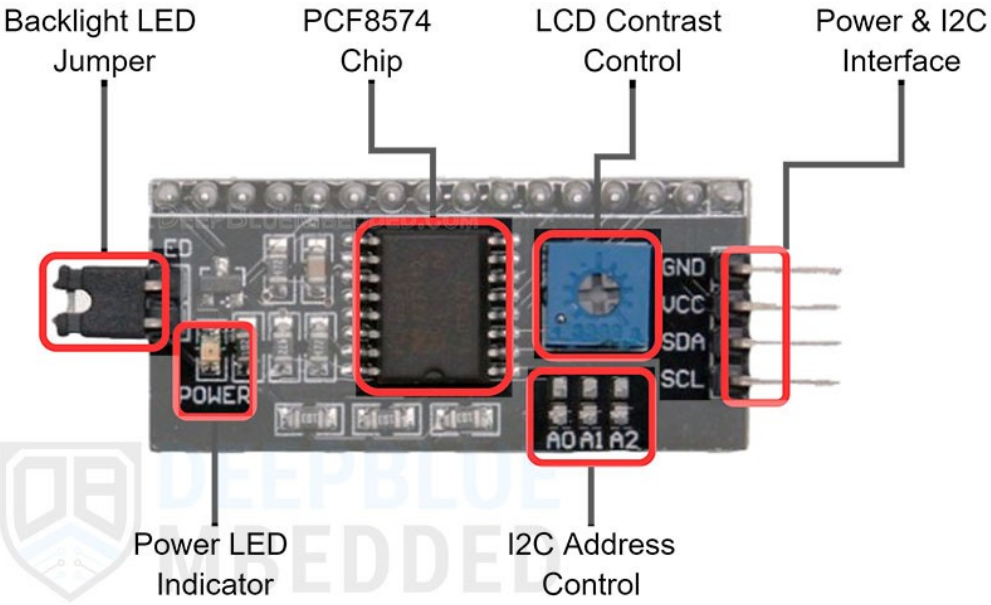


Fig 3.1 - I2C interface 20x4 LCD display.

1. I2C Wiring:

- **SDA (Serial Data):**

  Connect the LCD's SDA pin to the STM32F407's I2C SDA pin. You'll need to select the correct pin on your STM32F407 board. For example, on some boards, this might be PB10 or PB11.

- **SCL (Serial Clock):**

  Connect the LCD's SCL pin to the STM32F407's I2C SCL pin. Similarly, choose the appropriate pin on your board, which could be PB6 or PB7.

- **VCC (Power):**

  Connect the LCD's VCC pin to the STM32F407's 5V or 3.3V power supply, depending on the LCD's operating voltage.

- **GND (Ground):**
  Connect the LCD's GND pin to the STM32F407's ground pin.

2. I2C Address:

- The default I2C address for many I2C LCD modules is 0x27 or 0x3F. You might need to check your specific LCD module's datasheet for its exact address.
- This address is used by the STM32F407 to communicate with the LCD over the I2C bus.

3. STM32F407 Configuration:

- **I2C Library:**

  You'll need to include and configure the I2C library in your STM32F407 code.

- **Pin Configuration:**

  Make sure the chosen I2C pins are configured correctly in the STM32F407 HAL library (if you are using HAL) or using the relevant low-level I2C configuration functions.

- **Clock Speed:**
  Set the I2C clock speed appropriately. A common value is 100 kHz or 400 kHz.

4. Initialization:

- **Backlight:**

  The LCD's backlight is often controlled by a dedicated pin on the I2C interface. You'll need to find the appropriate pin (e.g., P3 on PCF8574) and set it to the desired state (ON or OFF) using the I2C commands.

- **LCD Initialization:**

You'll need to send the necessary initialization commands to the LCD over I2C to configure its display mode, font size, and other parameters.

5) MCP2551(CAN transreceiver)

The MCP2551 is a CAN bus transceiver that connects the CAN controller (like in the STM32F407) to the physical CAN bus.It converts the digital signals from the CAN controller to the differential voltages required for the CAN bus, and vice versa.The MCP2551 typically has a pinout that includes CANH, CANL, TXD, RXD, VDD, and GND.CANH and CANL are the differential outputs connected to the CAN bus.TXD is the transmitter output, and RXD is the receiver input, typically connected to the STM32F407's CAN controller.

CANH and CANL:

These are the differential outputs of the MCP2551. They connect to the CAN bus, which requires a 120 Ohm termination resistor at each end of the bus.

•TXD (Transmitter Data):

This pin is connected to the CAN transmitter output of the STM32F407. It's typically connected to the CAN_TD pin of the STM32F407, which is the data output.

•RXD (Receiver Data):

This pin is connected to the CAN receiver input of the STM32F407. It's typically connected to the CAN_RD pin of the STM32F407, which is the data input.

•VDD (Supply Voltage):

This pin is connected to the 3.3V or 5V power supply, depending on the MCP2551 variant (5V transceivers may require 5V supply, while 3.3V transceivers may operate at 3.3V).

•GND (Ground):

This pin is connected to the ground of the system.

•RS (Receive Select):
This pin is typically connected to a resistor, either 1kΩ or 10kΩ, to control the bus's rising/falling time, depending on the desired CAN bus speed and application.

- Implements ISO-11898 standard physical layer requirements
- Supports 1 Mb/s operation
- Suitable for 12V and 24V systems
- Externally-controlled slope for reduced RFI emissions
- Detection of ground fault (permanent dominant) on TXD input
- Power-on reset and voltage brown-out protection

• An unpowered node or brown-out event will not disturb the CAN bus
• Low current standby operation
• Protection against damage due to short-circuit conditions (positive or negative battery voltage)
• Protection against high-voltage transients
• Automatic thermal shutdown protection
• Up to 112 nodes can be connected
• High noise immunity due to differential bus implementation
• Temperature ranges: - Industrial (I): -40°C to +85°C - Extended (E): -40°C to +125°C

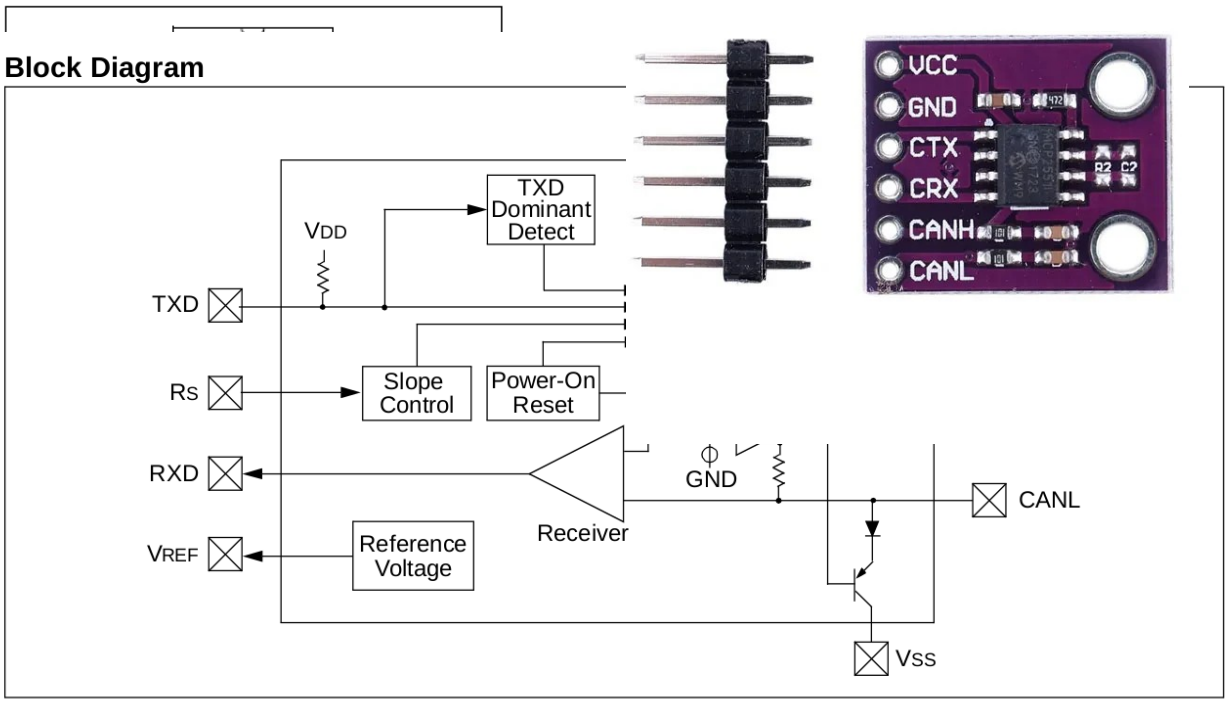


Fig 3.2 Block Diagram

Pin Number	Pin Name	Pin Function
1	TXD	Transmit Data Input
2	VSS	Ground
3	VDD	Supply Voltage
4	RXD	Receive Data Output
5	VREF	Reference Output Voltage
6	CANL	CAN Low-Level Voltage I/O
7	CANH	CAN High-Level Voltage I/O
8	RS	Slope-Control Input

4) Buzzer

This is an audio signaling device used at receivers side of system in this project.

Features :

- Diameter is 38mm Height is 22mm 2 Mounting holes distance: 40mm 2 wires.
- Rate voltage: 12v DC.
- Operating voltage: 3 – 24v
- Buzzer type: piezoelectric sound pressure level 95

Fig 4.1 – Buzzer

Software Requirements

1)STM32CubeIDE

Description

STM32CubeIDE is an all-in-one multi-OS development tool, which is part of the STM32Cube software ecosystem.

STM32CubeIDE is an advanced C/C++ development platform with peripheral configuration, code generation, code compilation, and debug features for STM32 microcontrollers and microprocessors. It is based on the Eclipse®/CDT™ framework and GCC toolchain for the development, and GDB for the debugging.

It allows the integration of the hundreds of existing plugins that complete the features of the Eclipse® IDE.

STM32CubeIDE integrates STM32 configuration and project creation functionalities from STM32CubeMX to offer all-in-one tool experience and save installation and development time. After the selection of an empty STM32 MCU or MPU, or preconfigured microcontroller or microprocessor from the selection of a board or the selection of an example, the project is created and initialization code generated. At any time during the development, the user can return to the initialization and configuration of the peripherals or middleware and regenerate the initialization code with no impact on the user code.

STM32CubeIDE includes build and stack analyzers that provide the user with useful information about project status and memory requirements.STM32CubeIDE also includes standard and advanced debugging features including views of CPU core registers, memories, and peripheral registers, as well as live variable watch, Serial Wire Viewer interface, or fault analyzer.

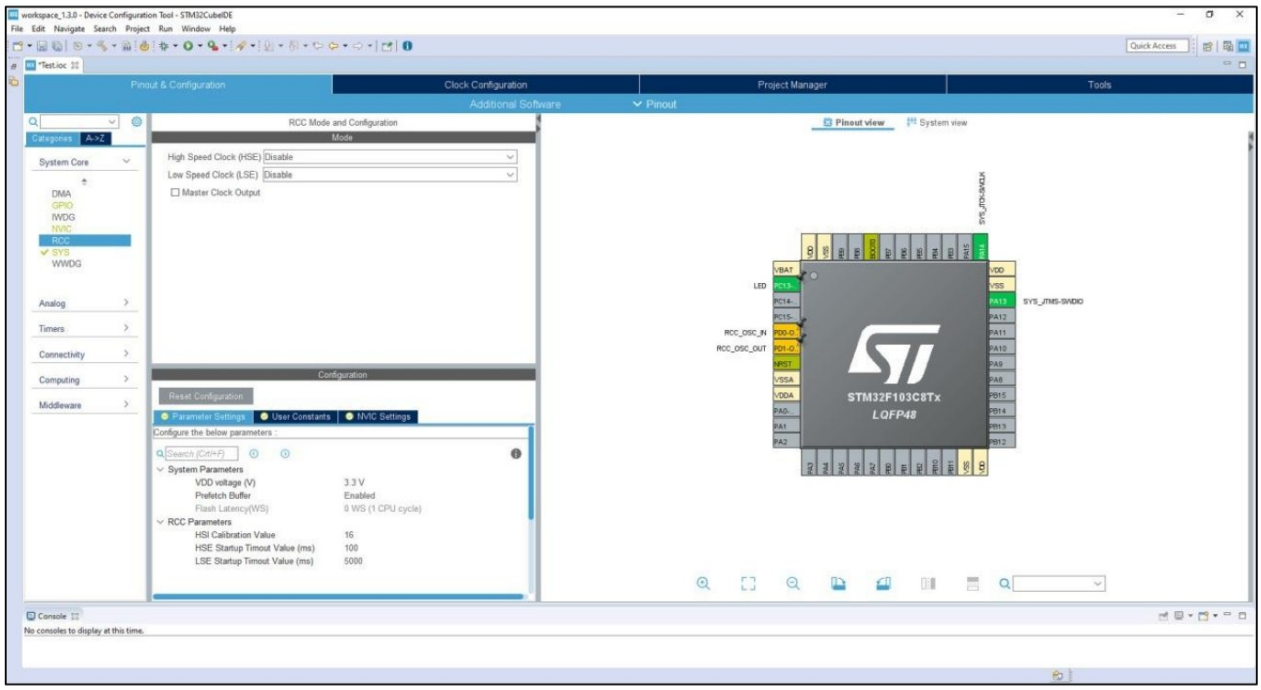


fig 1.1. STM32 cube IDE interface

All features:

- Integration of services from STM32CubeMX:
- Based on Eclipse /CDT , with support for Eclipse add-ons, GNU C/C++ for Arm toolchain and GDB debugger
- STM32MP1 Series:
- Additional advanced debug features including:
- Support for ST-LINK (STMicroelectronics) and J-Link (SEGGER) debug probes
- Import project from Atollic TrueSTUDIO and AC6 System Workbench for STM32 (SW4STM32)
- Multi-OS support: Windows , Linux , and macOS , 64-bit versions only