INTERNSHIP REPORT

NITHIKA A

BTECH CSE (IOT)

(05-06-24 TO 12-06-24)

SKILLS LEARNT

- Induction motor efficiency monitoring system software
  -.NET (winform desktop application)(C#)
  -UI/UX
  -MODBUS (RS485)
  -EM6400 schneider display
  -basic hardware
- Module: Instrumentation
- TIME event host cum volunteer
- Nextion display
- Basic electronics

INDUCTION MOTOR EFFICIENCY MONITORING SYSTEM (IMEMS) - SOFTWARE

BASIC HARDWARE:

The major components of the system are

- Auto transformer
- Control panel
- Test bed-1 (15-18 Hb torque)
- Test bed-2 (50-75 Hb torque)
- Test bed-3 (5-8 Hb torque)
- DAC – data acquisition control

Aim: to calculate the efficiency of any given motor.

Procedure:

EFFICIENCY = (OUTPUT/INPUT)

There is no direct way to calculate the output. However it can be obtained from the input as

OUTPUT = INPUT - LOSSES

We can identify the measure of losses such as hysteresis, eddy current, bearing losses.

The losses are given by the thermodynamic principles.

$$\text{INPUT POWER} = \sqrt{3} V_L I_L \cos\Phi$$

$$\text{OUTPUT POWER} = (2\pi 60)nT$$

Target:

- ➔ To develop a software inspired from the modscan (labview medium) using the dotnet framework (language: C#)
- ➔ Develop a interactive and creative UI/UX
- ➔ Communicate using the modbus protocol for the schneider display and proprietary protocol for the native display
- ➔ Simulate the relay to turn it on and off using the magnetic flux principle

DOTNET FRAMEWORK:

The .NET framework was developed by Microsoft, is a comprehensive software framework used for building and running applications on Windows. When creating forms for desktop applications in C#, one commonly uses Windows Forms (WinForms) and Windows Presentation Foundation (WPF).

Advantages:

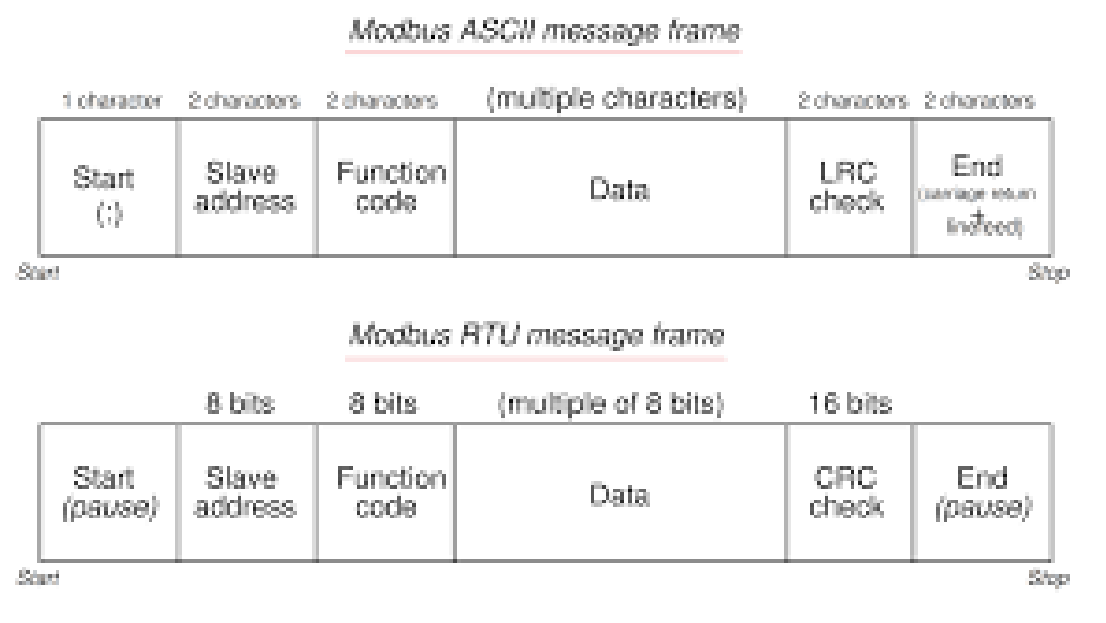
Base Class Library (BCL) is part of the large class library (FCL) and provides classes, interfaces, and value types that expedite the development of applications.

IDE: Microsoft Visual Studio

MODBUS PROTOCOL:

It works on the master slave communication where one master takes control over all the slaves to process informations such as line voltage, line current, average power factor, frequency, total input power.

Framework:



Details:

Input of start address should be 1 less than the actual start address.

Slave address (point type): = master -> normal response

= master+80 -> exceptional

Choosing the slave address was based on the EM6400 display (via datasheet) devised by the Schneider and the necessary data to be read are:

- Line Voltage – [3909]
- Line Current – [3913]
- Average Power factor – [3907]
- Frequency – [3915]
- Total input power – [3903]

It communicates via the RS485 channel and is connected to serial port.

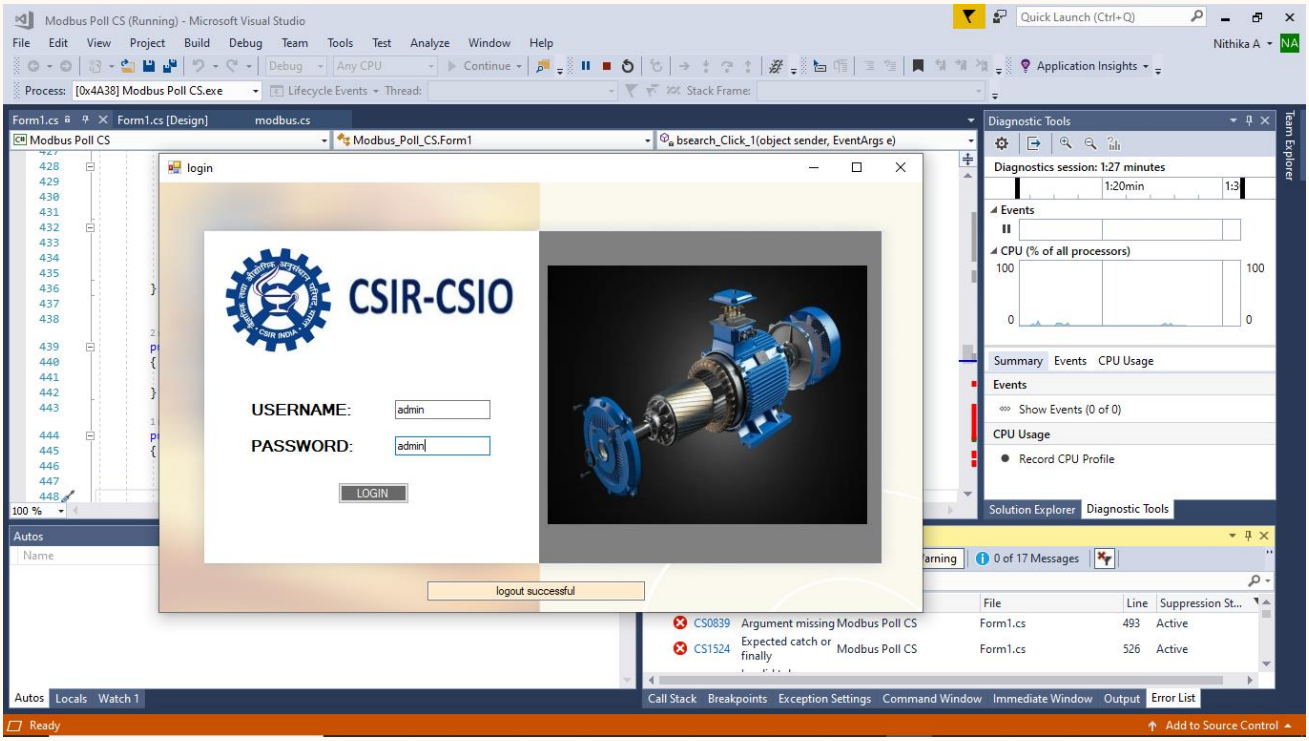
Connected via the bus topology.

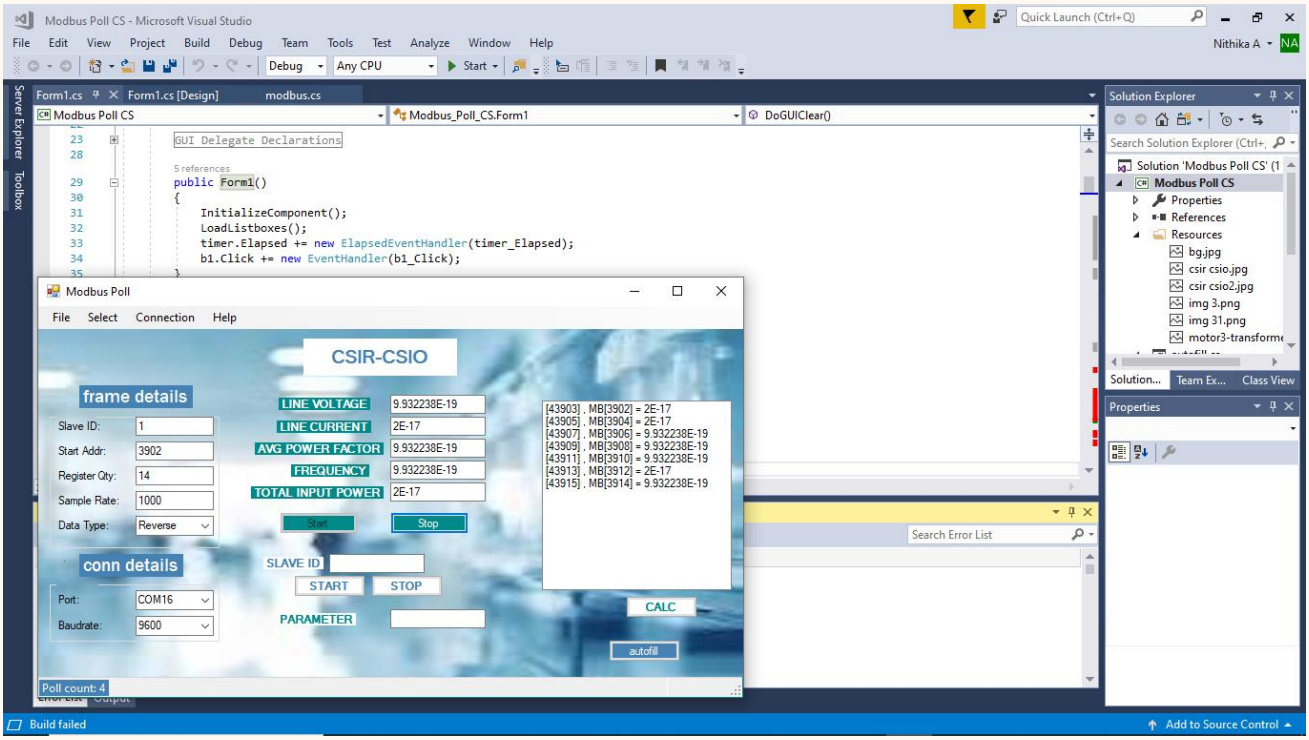
PROPRIETARY PROTOCOL:

It behaves in such a way that when a character is sent via the serial port, its corresponding data gets displayed if it is connected on the same baud rate (9600).

This is taken care of by the data acquisition control.

OUTPUT:


Modbus Poll CS (Running) - Microsoft Visual Studio
login
CSIR-CSIO
USERNAME:
admin
PASSWORD:
admin
LOGIN
logout successful
Diagnostic Tools
Diagnostics session: 1:27 minutes
Events
CPU (% of all processors)
Summary
Events
CPU Usage
Show Events (0 of 0)
Record CPU Profile
Solution Explorer
Diagnostic Tools
CS0839 Argument missing Modbus Poll CS Form1.cs 493 Active
CS1524 Expected catch or finally Modbus Poll CS Form1.cs 526 Active
Call Stack
Breakpoints
Exception Settings
Command Window
Immediate Window
Output
Error List
Ready
Add to Source Control


Modbus Poll CS - Microsoft Visual Studio
Modbus Poll
File
Select
Connection
Help
CSIR-CSIO
frame details
Slave ID: 1
Start Addr: 3902
Register Qty: 14
Sample Rate: 1000
Data Type: Reverse
conn details
Port: COM16
Baudrate: 9600
LINE VOLTAGE 9.932238E-19
LINE CURRENT 2E-17
AVG POWER FACTOR 9.932238E-19
FREQUENCY 9.932238E-19
TOTAL INPUT POWER 2E-17
Start
Stop
SLAVE ID
START
STOP
PARAMETER
CALC
autofill
[43903] , MB[3902] = 2E-17
[43905] , MB[3904] = 2E-17
[43907] , MB[3906] = 9.932238E-19
[43909] , MB[3908] = 9.932238E-19
[43911] , MB[3910] = 9.932238E-19
[43913] , MB[3912] = 2E-17
[43915] , MB[3914] = 9.932238E-19
Poll count: 4
Solution Explorer
Modbus Poll CS
Properties
References
Resources
bg.jpg
csir csio.jpg
csir csio2.jpg
img 3.png
img 31.png
Properties
Build failed
Add to Source Control

DEVICE:


IEICOS MOTOR TEST BENCH COMPUTERISED
IEICOS
MAKE : IEICOS
MODEL:IMTB
INDUSTRIAL ENGINEERING INSTRUMENTS
203, 12TH MAIN, 3RD PHASE, PEENYA,
BANGALORE - 560058
PH NO :-91-80-28394520
00066
00067
00069
00070
108.5
rtd

GROUP 1: (MODBUS)

S NO	INSTRUMENT	SLAVE ADDR
1	SCHNEIDER (TEST BED 1&3)	01
2	SCHNEIDER (TEST BED 2)	02

GROUP 2: (PROPREITARY)

S NO	SENSOR	SLAVE ADDR
1	SPEED TEST(1&3)	A
2	REACTION TORQUE (1)	B
3	LOCKED ROTOR TORQUE (1&3)	C
4	SPEED(2)	R
5	REACTION TORQUE (2)	J
6	LOCKED ROTOR TORQUE(2)	E
7	REACTION TORQUE (3)	G
8	LOCKED TORQUE (3)	H
9	4CH TEMP INDICATOR	03

PARAMETERS:

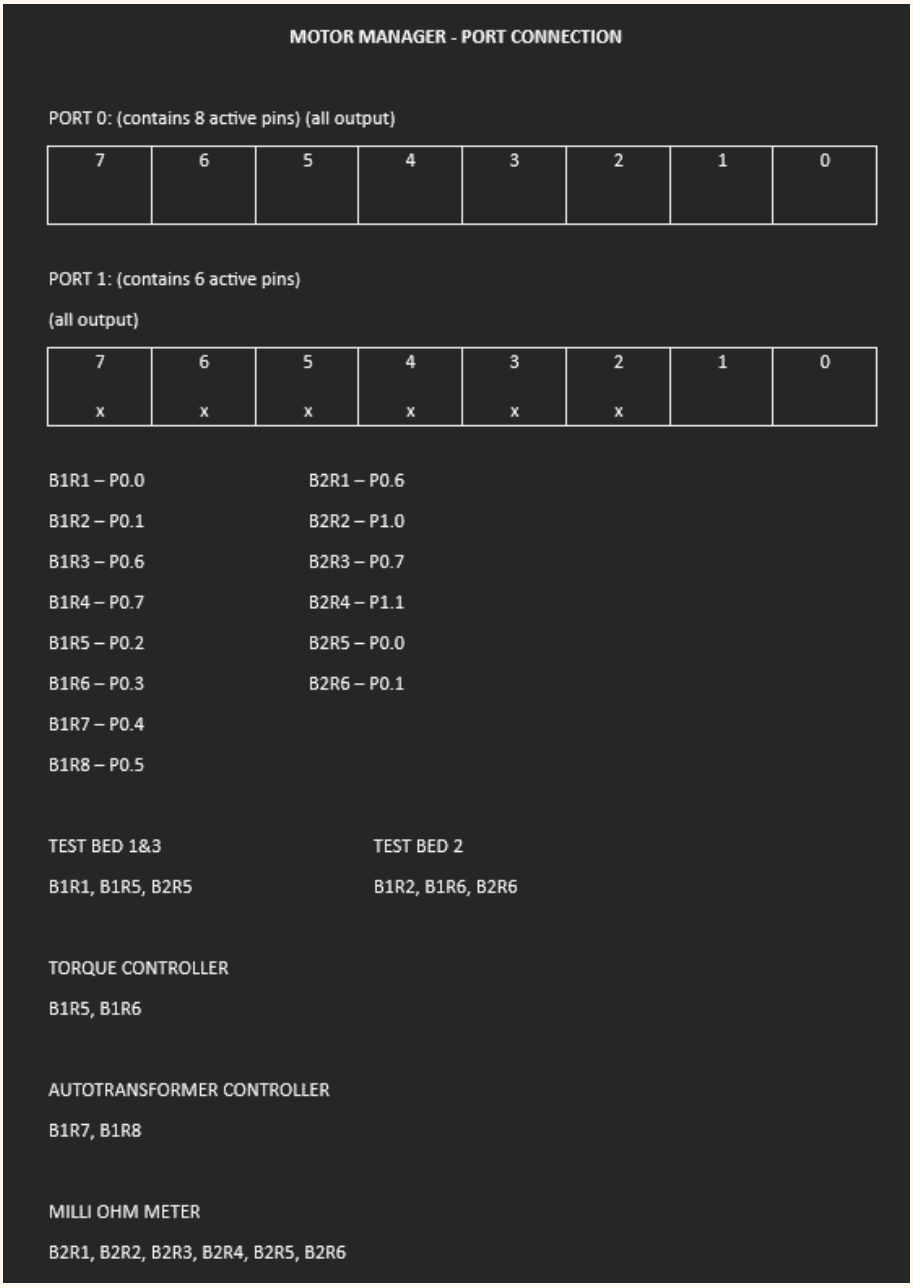
Baud rate – 9600

Data bits – 8

Parity – none

Flow control – none

SIMULATION:



MOTOR MANAGER - PORT CONNECTION

PORT 0: (contains 8 active pins) (all output)

7	6	5	4	3	2	1	0

PORT 1: (contains 6 active pins)

(all output)

7	6	5	4	3	2	1	0
x	x	x	x	x	x		

B1R1 – P0.0
B1R2 – P0.1
B1R3 – P0.6
B1R4 – P0.7
B1R5 – P0.2
B1R6 – P0.3
B1R7 – P0.4
B1R8 – P0.5

B2R1 – P0.6
B2R2 – P1.0
B2R3 – P0.7
B2R4 – P1.1
B2R5 – P0.0
B2R6 – P0.1

TEST BED 1&3
B1R1, B1R5, B2R5

TEST BED 2
B1R2, B1R6, B2R6

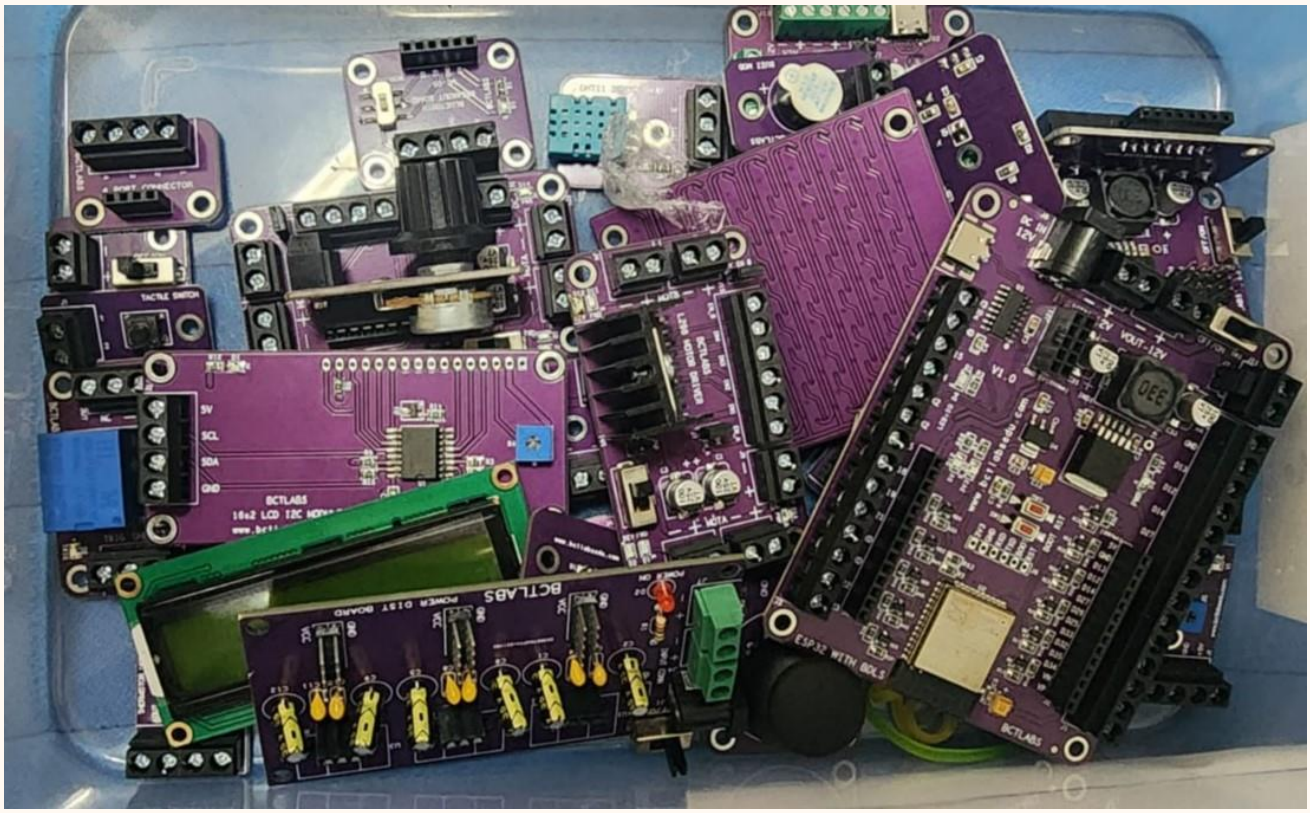
TORQUE CONTROLLER
B1R5, B1R6

AUTOTRANSFORMER CONTROLLER
B1R7, B1R8

MILLI OHM METER
B2R1, B2R2, B2R3, B2R4, B2R5, B2R6

BASIC ELECTRONICS

Learnt introduction to basic components such as a LDR (light dependent resistor), RTD (resistance temperature detector), servo motor and others.



Introduction and working with basic IoT components in the Arduino workspace.

Overview to the solar energy harvester (SEH) and Hydrogen leakage detection system.

MODULE : INSTRUMENTATION

TOPICS:

->Transformer

->Transistors

->components of instrumentation

- ->Sensor
- ->Processor/controller
- ->Accelery output
- ->Power supply
- ->Input devices

->Sensors and characteristics

->Process of instruments

- ->Measuring
- ->Processing
- ->Controlling
- ->Communicating
- ->Automating

TIME AND SSM - VOLUNTEER

TIME (Technologist Industrialist Meet and Expo):

Got the chance to explain the air quality monitoring system (previo and present it to various industrialists in the stall that was maintained by Susima Technologies.

Explained the motor lab and its working to various scientists and the software, its motive and working.

Attended lectures related to the one week one theme – energy

Gathered a lot of information regarding the upcoming technologies.

SSM (Student Scientist Meet):

Hosted, presented, volunteered and helped in various events during the meet where students from various schools came to interact and ask questions to scientists.

8 CHANNEL TEMPERATURE DATALOGGER – USING NEXTION DISPLAY

DATA LOGGER:

A data logger is an electronic device designed to automatically record data over time or in response to specific events. It typically includes a built-in or connected sensor, a microprocessor, and memory for storing data. Data loggers are used in various fields such as environmental monitoring, industrial automation, healthcare, and research to collect and store data for analysis and decision-making.

Key Components of a Data Logger:

Sensors:

Measure specific parameters like temperature, humidity, pressure, voltage, or current.

Microprocessor:

Processes the input data from the sensors and controls the data logging process.

Memory:

Stores the recorded data, which can be in the form of internal memory or external storage like SD cards or USB drives.

Power Source:

Can be battery-powered, solar-powered, or connected to an external power supply.

Communication Interface:

Allows data transfer to other devices or systems, often via USB, Bluetooth, Wi-Fi, or serial ports.

Software:

Used for configuring the data logger, viewing, analyzing, and exporting the recorded data.

NEXTION:

Nextion is a human machine interface (HMI) solution combining an onboard processor and memory touch display with Nextion Editor software for HMI GUI project development.

Main aspects of nextion:

• Display Types: Nextion displays come in various sizes from 2.4" to 7.0", offering touch-sensitive screens for interactive applications.

• Touch Sensitivity: Capacitive touch screens enable direct interaction with graphical elements using fingers or styluses.

• Resolution: Supports resolutions ranging from 320x240 to 800x480 pixels, ensuring clear and detailed graphics.

• Communication Interfaces: Supports UART communication, compatible with Arduino, Raspberry Pi, and other microcontroller platforms.

• User Interface Design: Customizable elements like buttons, sliders, and text boxes with touch-responsive capabilities.

• Memory and Storage: Onboard Flash memory and microSD card support for storing GUI designs and data.

• Power Requirements: Operates on 5V DC power supply, suitable for both battery powered and mains-powered applications.

CREATING USER INTERFACE

Creating an 8-channel temperature data logger requires several key components to ensure accurate data acquisition and storage.

1. Microcontroller: Central unit managing data acquisition from temperature sensors and controlling data storage.

2. Temperature Sensors: Devices measuring temperature at each channel accurately.

3. Analog-to-Digital Converter (ADC): Converts analog temperature sensor outputs into digital signals for processing.

4. Storage Medium: Stores temperature data logs, typically using Flash memory, EEPROM, or SD card modules.

5. Real-Time Clock (RTC): Provides accurate timestamps for logged temperature readings.

6. Power Supply: Provides reliable power to ensure continuous operation of the data logger.

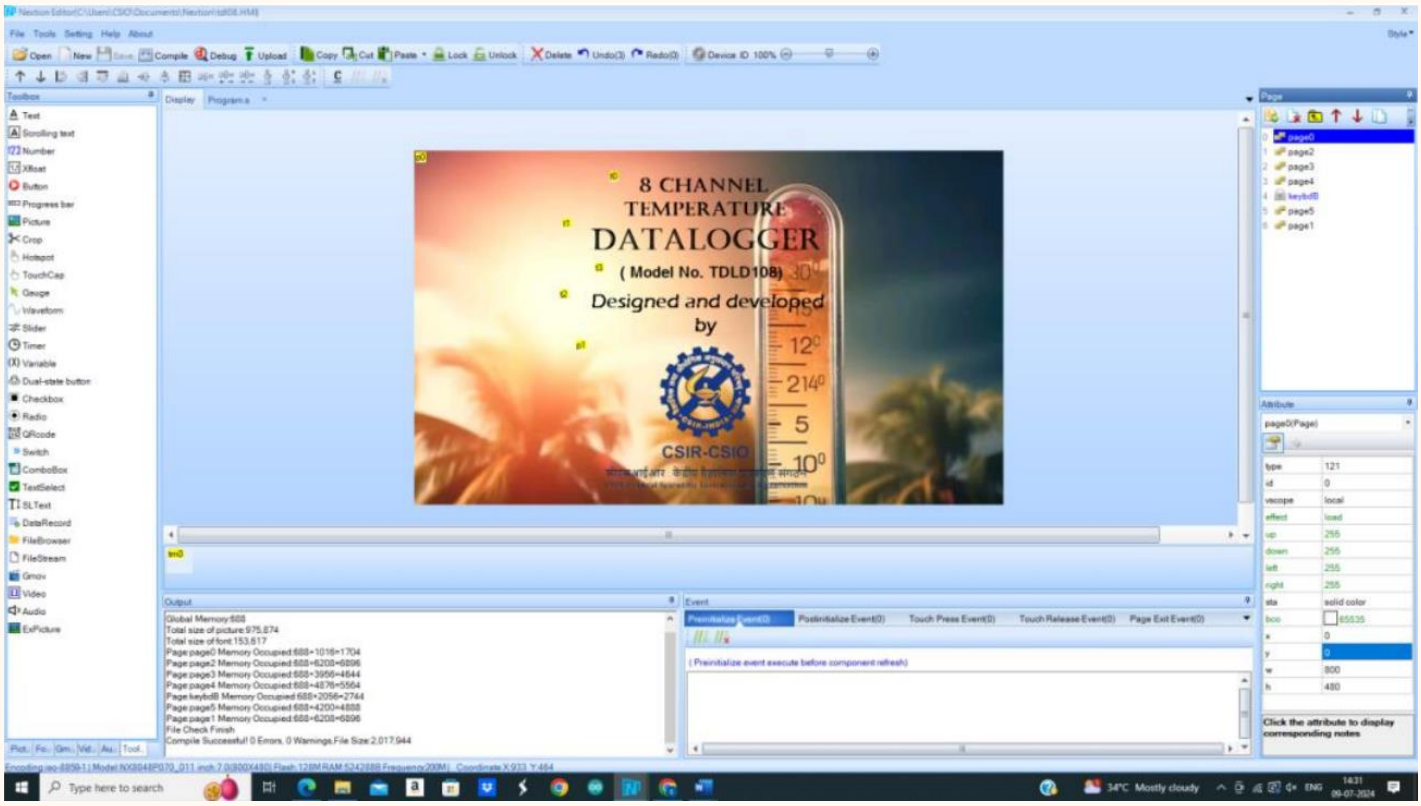
7. Display (Optional): Shows real-time temperature readings or system status for monitoring purposes.

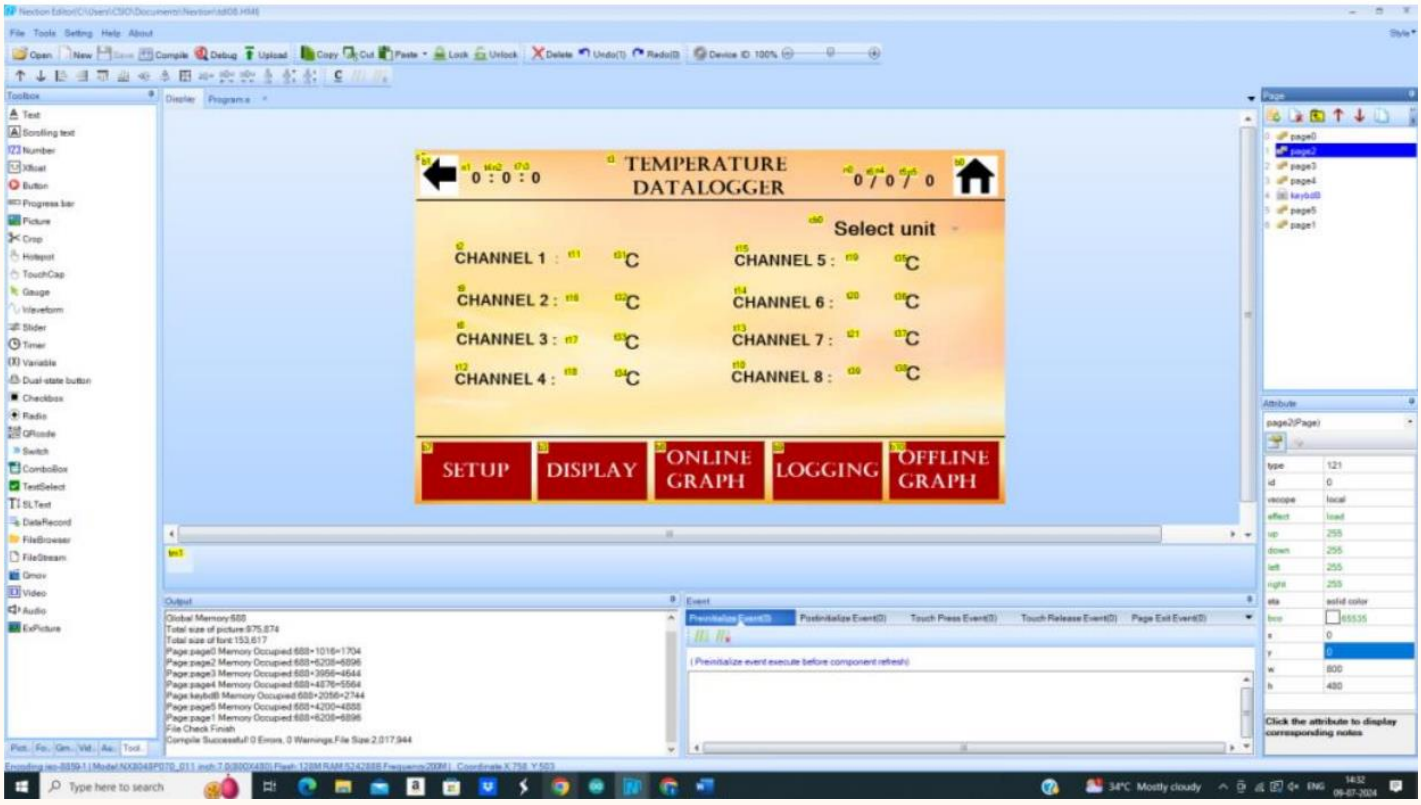
8. Enclosure and Mounting: Protects components and ensures accurate sensor placement for reliable temperature measurement.

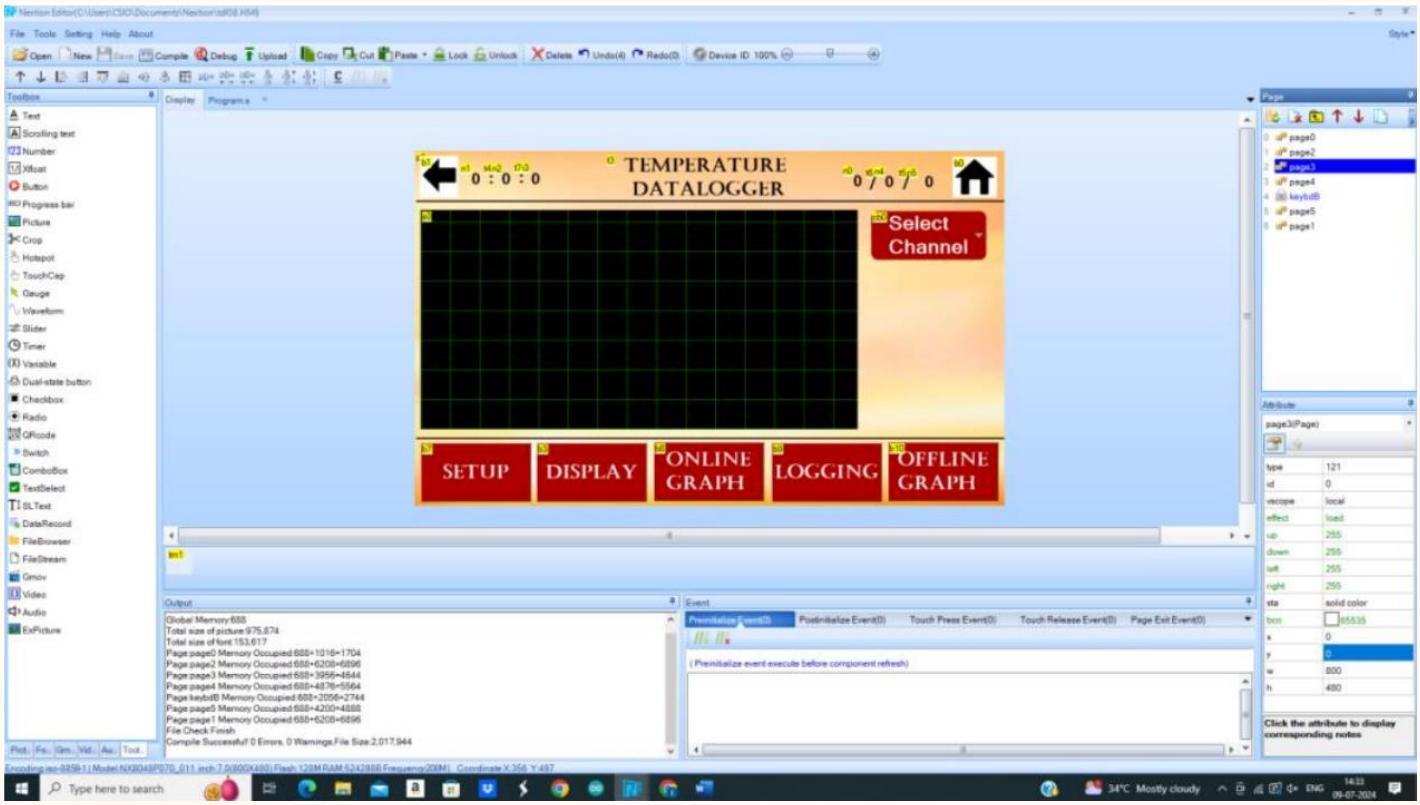
9. Communication Interface (Optional): Allows data transfer or real-time monitoring via UART, Wi-Fi, Ethernet, etc.

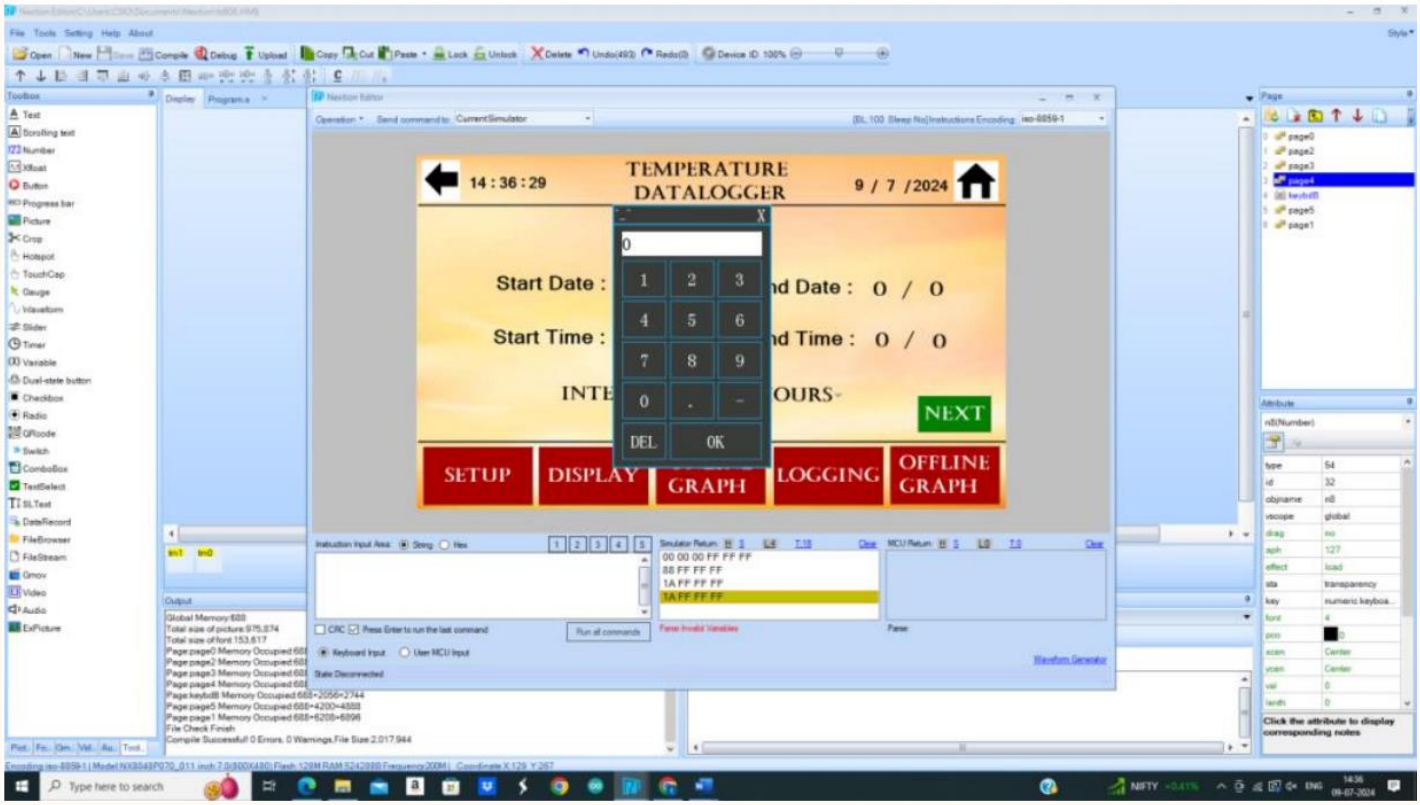
10. Programming Environment: Software environment for coding and uploading firmware to the microcontroller, such as Arduino IDE.

OUTPUT:


8 CHANNEL
TEMPERATURE
DATALOGGER
(Model No. TDLD108)
Designed and developed
by
CSIR-CSIO


TEMPERATURE
DATALOGGER
Select unit
CHANNEL 1 :
CHANNEL 2 :
CHANNEL 3 :
CHANNEL 4 :
CHANNEL 5 :
CHANNEL 6 :
CHANNEL 7 :
CHANNEL 8 :
SETUP
DISPLAY
ONLINE GRAPH
LOGGING
OFFLINE GRAPH

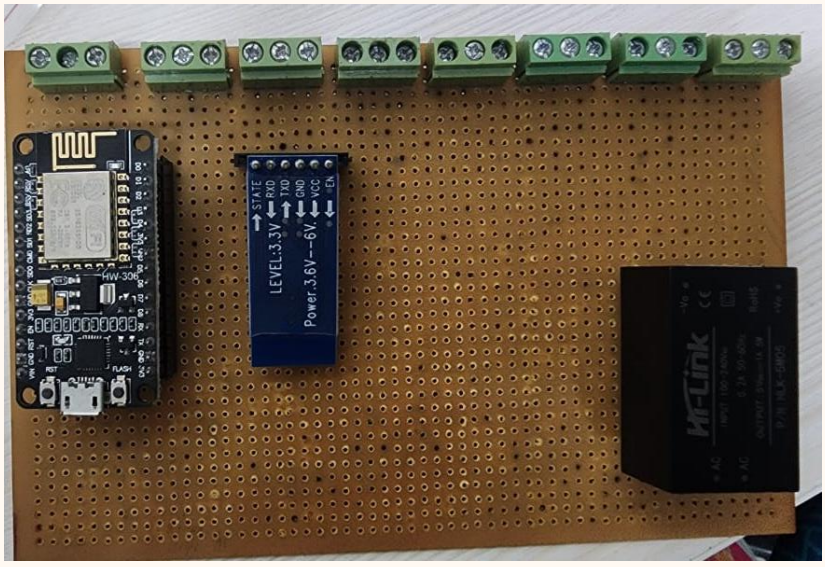

TEMPERATURE
DATALOGGER
Select
Channel
SETUP
DISPLAY
ONLINE
GRAPH
LOGGING
OFFLINE
GRAPH


14 : 36 : 29
TEMPERATURE
DATALOGGER
9 / 7 /2024
Start Date :
Start Time :
NEXT
SETUP
DISPLAY
GRAPH
LOGGING
OFFLINE
GRAPH

WORKING:

- ➔ A splash screen has been set up to display the company's name and related details (time displayed: 5s -> 5000 milli sec)
- ➔ It then enters into the home screen where all the temperature data gets displayed to denote the smooth functioning.
- ➔ By navigating to the 'logging' screen, we will be able to set the start and end time as well as the interval in which it has to be displayed.
- ➔ The logged data can be retrieved later.
- ➔ Both online and offline graphs can be viewed.

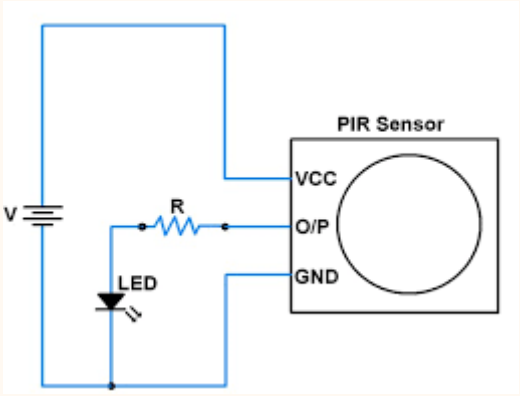
HARDWARE:



Worked on soldering and assembling various components such as 3 pin connectors, HC-05 module, NodeMCU, power supply and others.

HOME AUTOMATION SYSTEM USING INFRARED MOTION SENSOR

Connection diagram:



PIR sensors utilise the detection of infrared that is radiated from all objects that emit heat. They sense the presence of people, animals and objects through the movement of their infrared wavelength.