**DOSING CONTROLLER**

**Version : 3**

**Date Updated : 5 - 5 - 2022**

**Overview :**

Dosing Controller is used for greenhouse automation where it controls the growth of plants taking input from sensors like : ph, ec, do, water level sensor and controls the working of submersible pump, heater, mixer and chamber.

Meanwhile it also sends data collected from the sensors based on user interest and sends it to the cloud via controller.

It works on three modes : Manual mode, Schedule Mode and Auto Mode. Where users can select any of these to control as per needs.

To connect to wifi for transfer of data collected it needs to connect to the controller. via the input from Nextion device ( wifi page ).

**Recommended System Requirements :**

The Dosing Controller works taking input from Nextion Device and Sends the data to Arduino and in some cases for wifi setup and to send data to cloud it needs in build esp - 8266 to connect to arduino.

The dosing controller coding is done through Arduino IDE .The software is open source and can be installed from the Arduino website.

Website : <https://www.arduino.cc/>

The Nextion device consists of purely C language coding within the software provided by Nextion manufacturer. The file with nextion software supported extension is given below. The Software is open source and can be installed from the Nextion website.

Website : <https://nextion.tech/>

The UI of nextion device is designed using the Inkscape software. The file with the inkscape supported extension is given below. The Software is open source and can be installed from the Inkscape website.

Website : <https://inkscape.org/>

**Sample Code :**

There are two communication here :

1. From Nextion to Arduino : For this purpose The function

**func\_name(){**

**……….**

**………..**

**……………**

**}**

1. From Arduino To Nextion : For this purpose the function

**func\_name()**

**{ …………**

**…………**

**…………} is used.**

**How to use the Software :**

The basic things required for this software to use are:

1. UI design
2. Pure C Language Coding

Below I’m sharing the Youtube Channel from where one can easily understand the project.

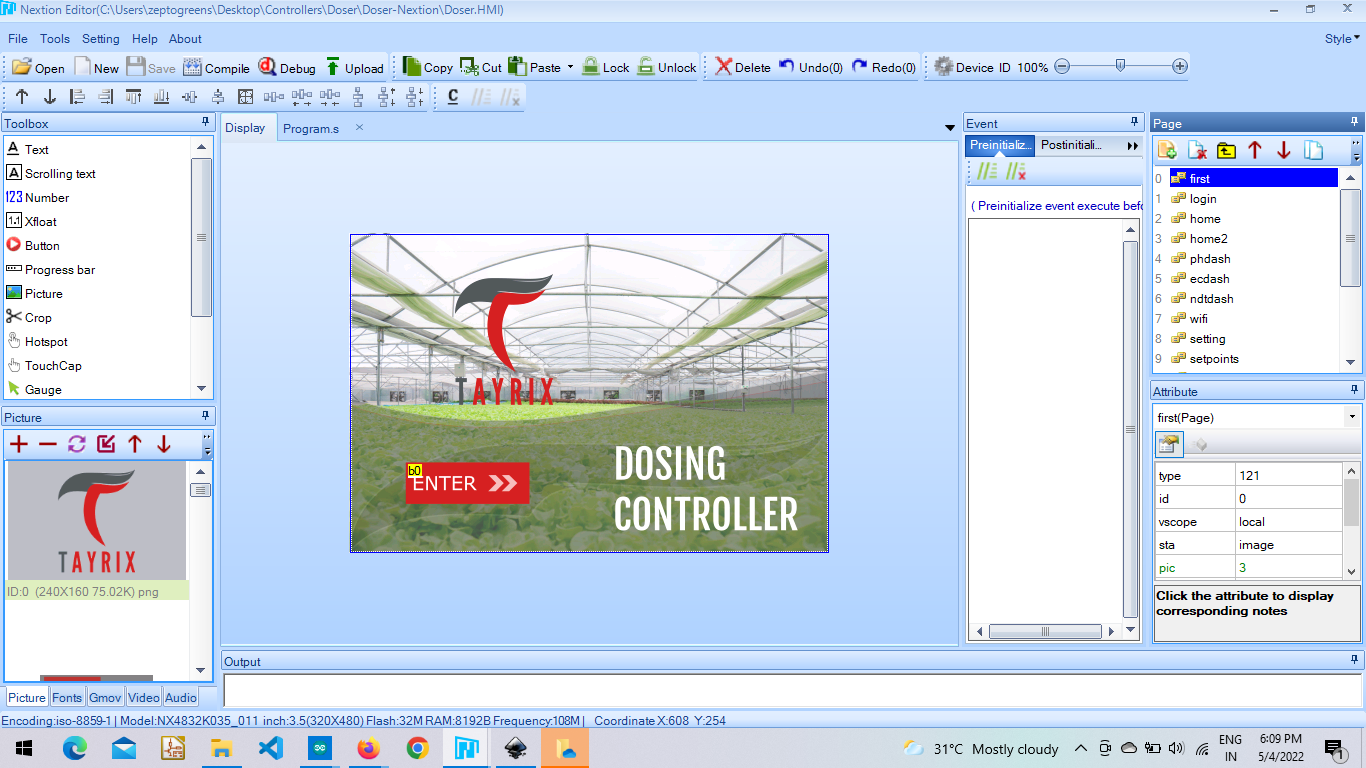
Everything in the project has taken reference from the videos of this channel only.

Reference : <https://www.youtube.com/c/CheapControls>

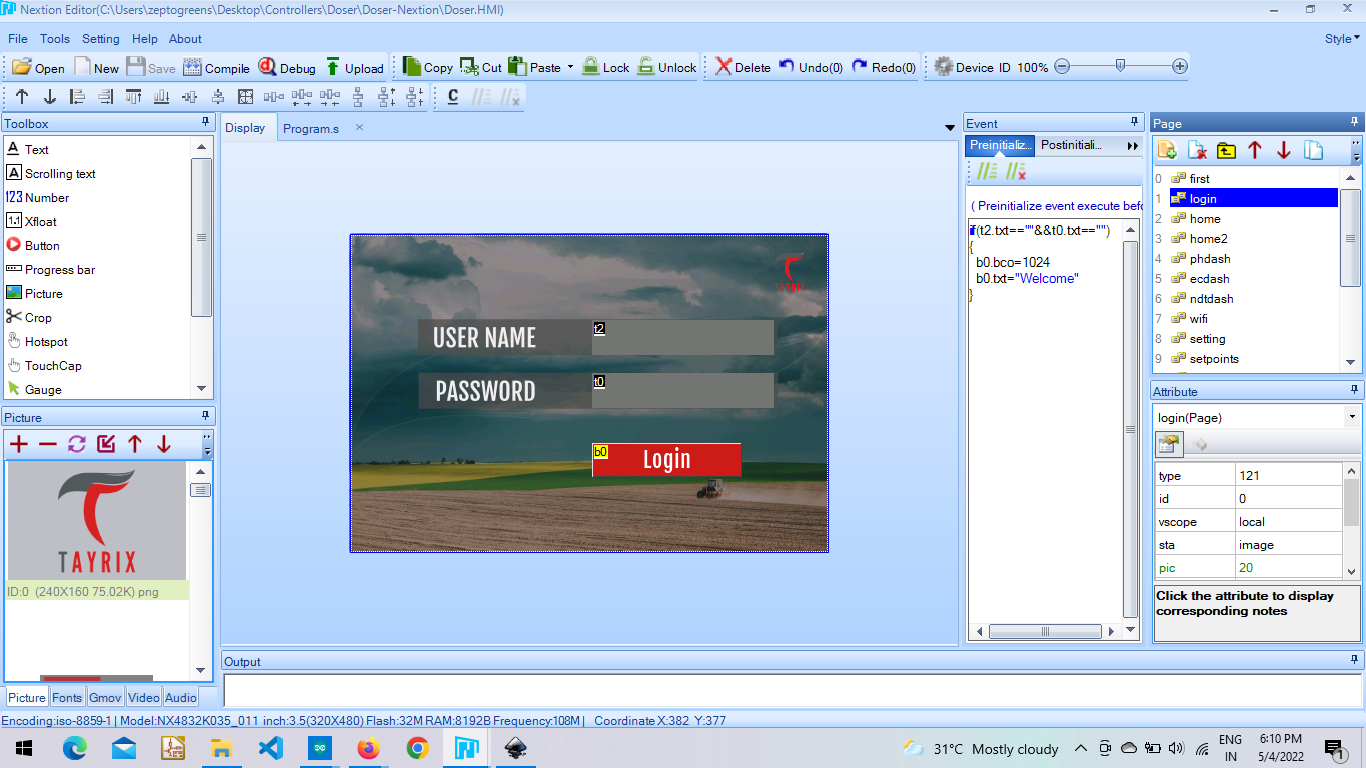
**Summary of Project :**

This project consists of a total of 38 pages of Nextion.

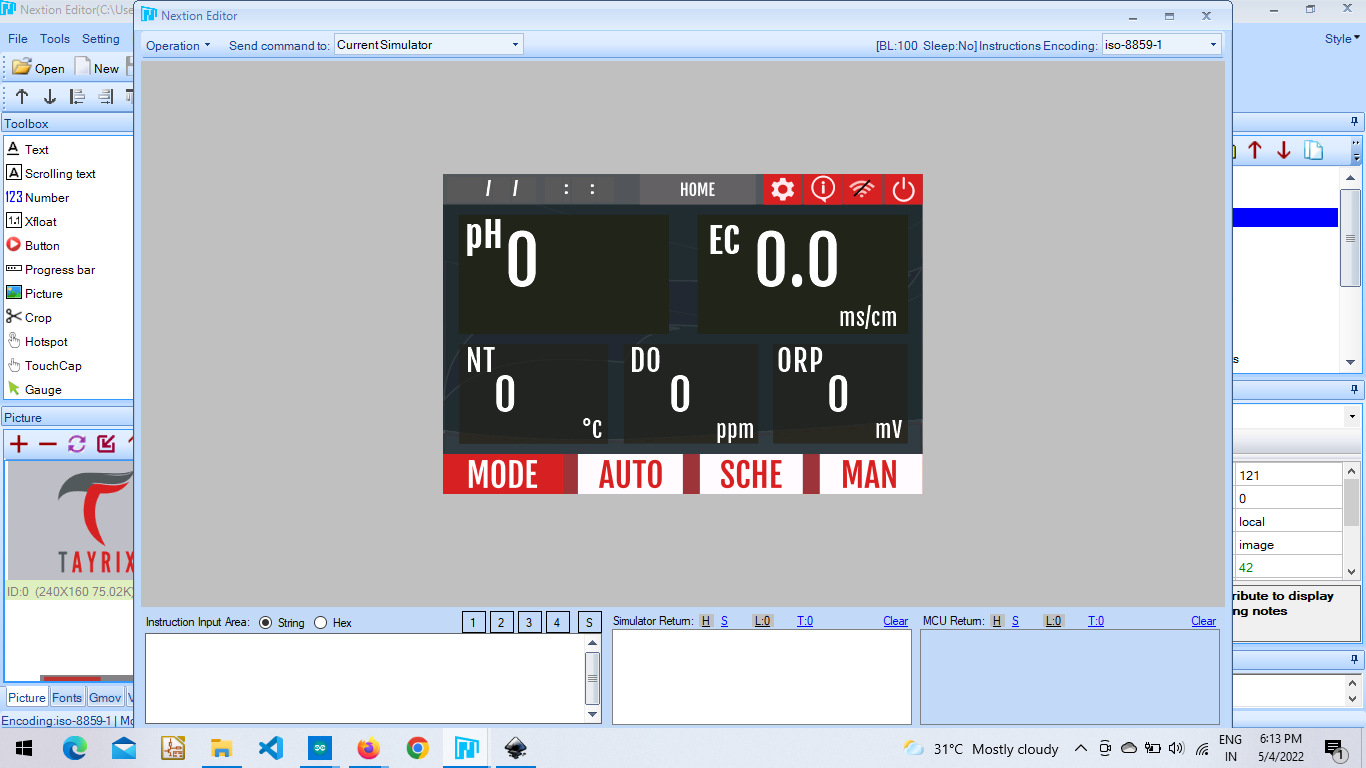
Welcome page



Login Page.



**Home Page :** Consists of the input showing from sensors such as : pH sensor, Ec sensor, Water Temperature sensor, Do sensor, ORP sensor. And navigation buttons for respective modes like: Auto mode, Schedule and Manual mode.



**func\_name(){**

**func\_name(){**

**………..**

**………..**

**………..**

**}**

**func\_name(){**

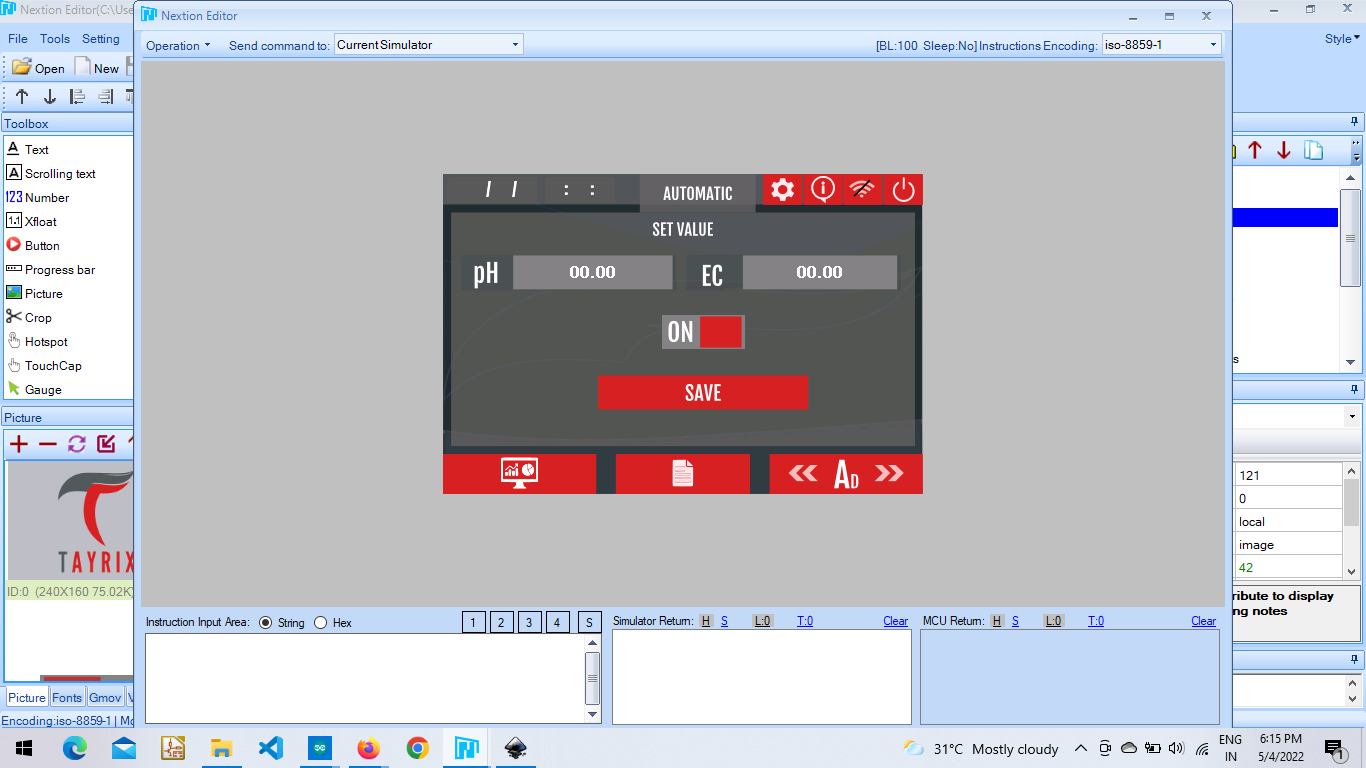
**………**

**………**

**………**

**}**

**Auto Mode Page :** It is used to set the required ph and ec value to be in the water running for plants. The toggle button here is ON/OFF. After entering the values the user needs to save them and then turn ON to run it.

****

**func\_name(){**

**func\_name(){**

**………..**

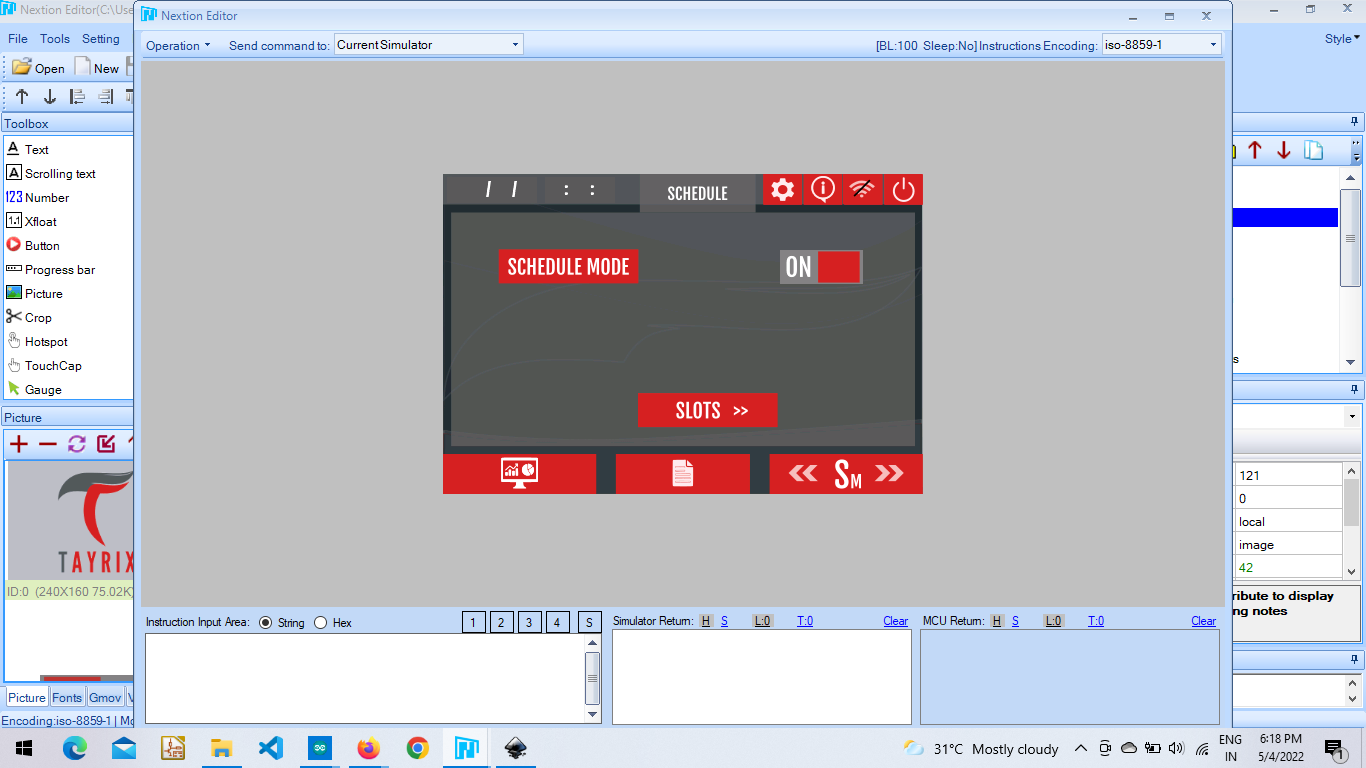
**………..**

**………..**

**}**

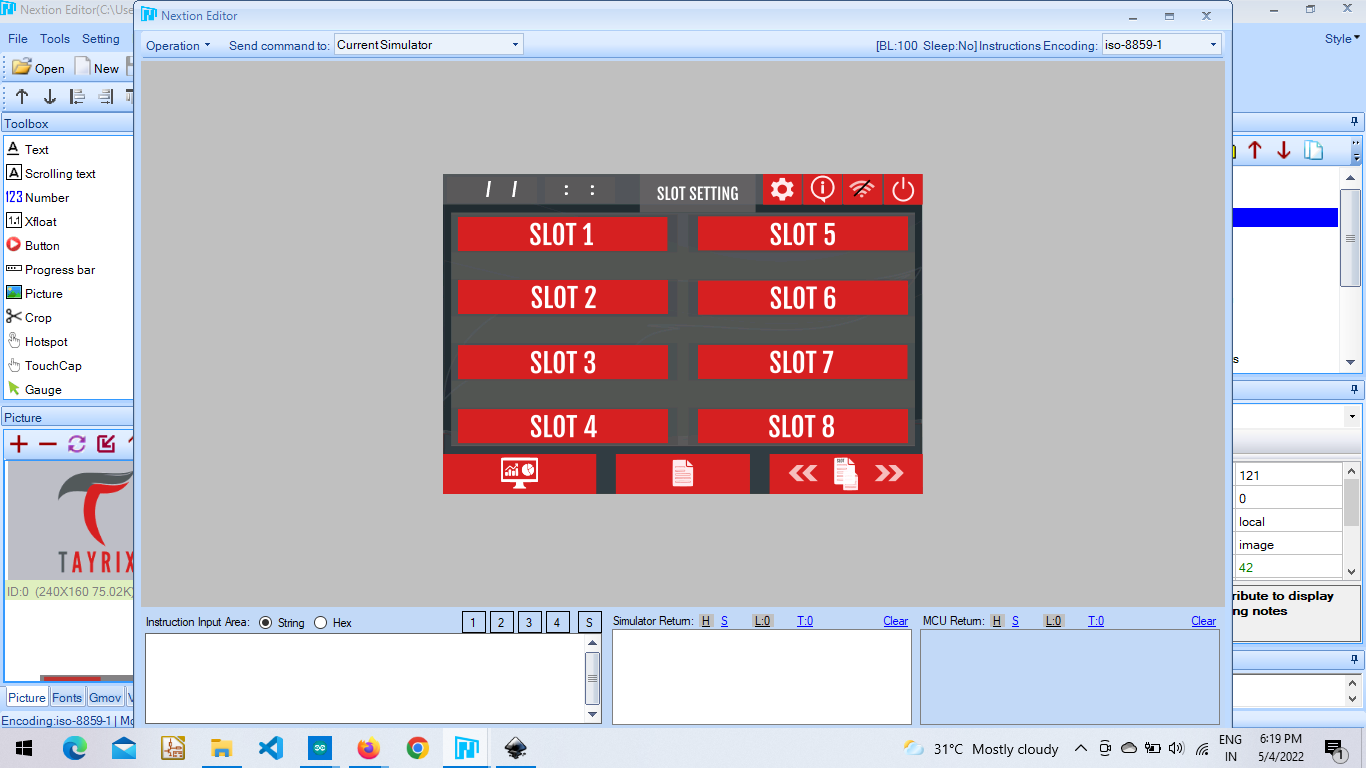
**Schedule Mode Page :** This page is for ON/OFF the schedule mode after entering the values in the given slots.

There is a code return when we click the toggle button.

****

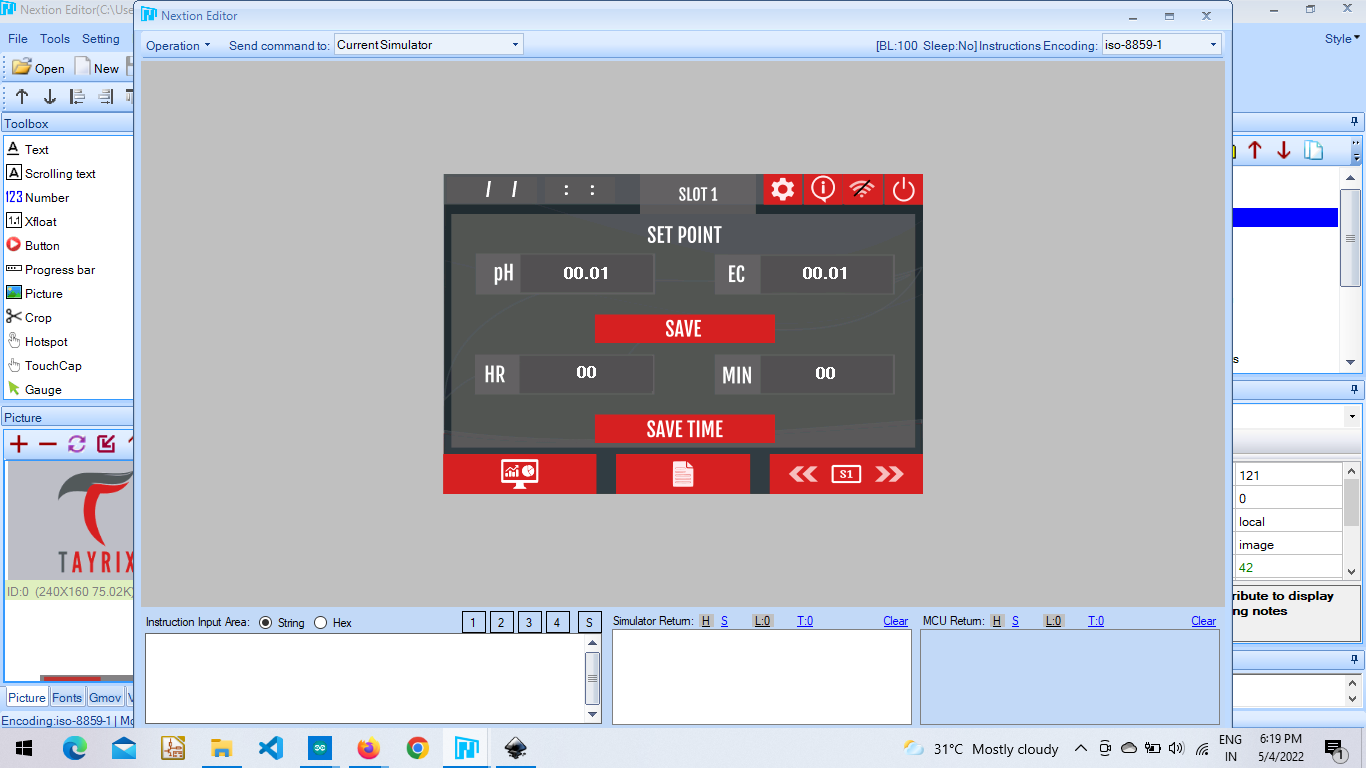
This page is to select slots depending on the need. It consists of only 8 slots.

There is a code return on this page for indication purposes whether the slot is active or not.

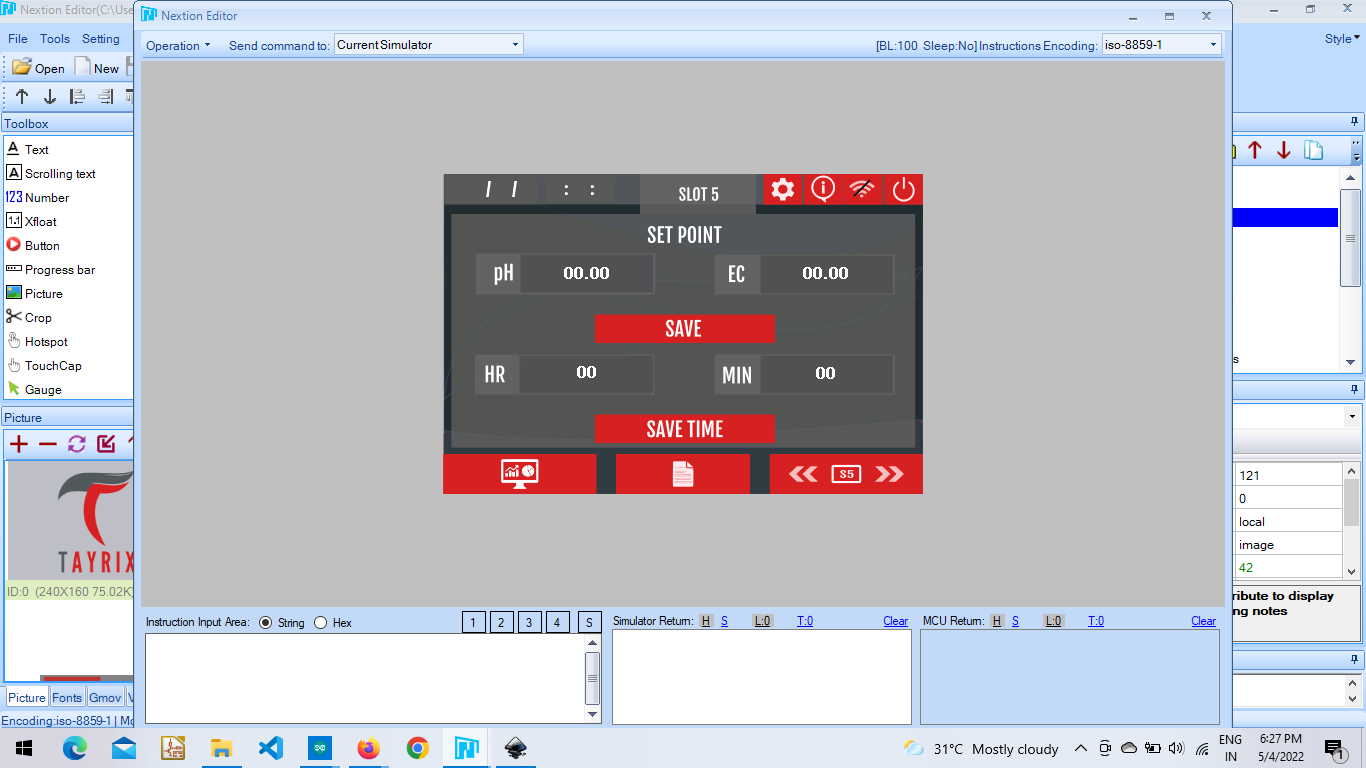
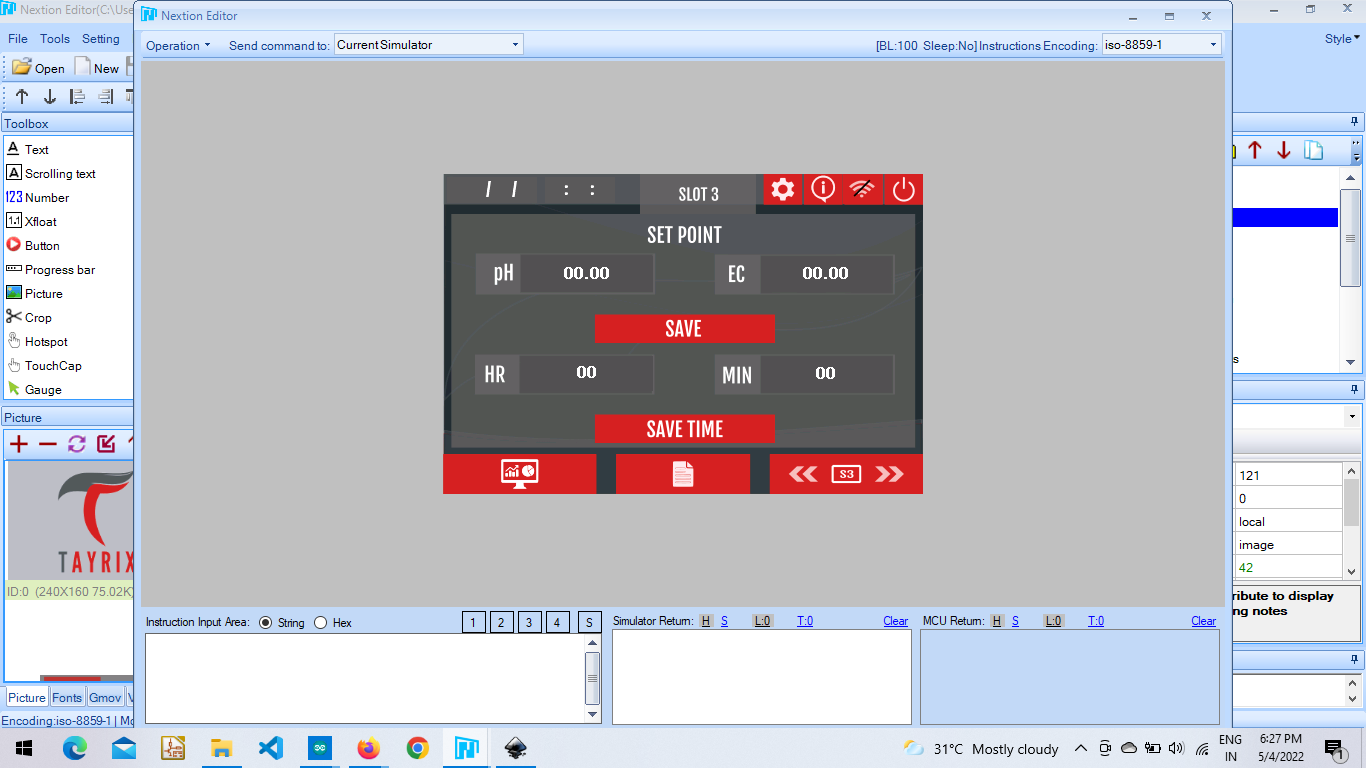
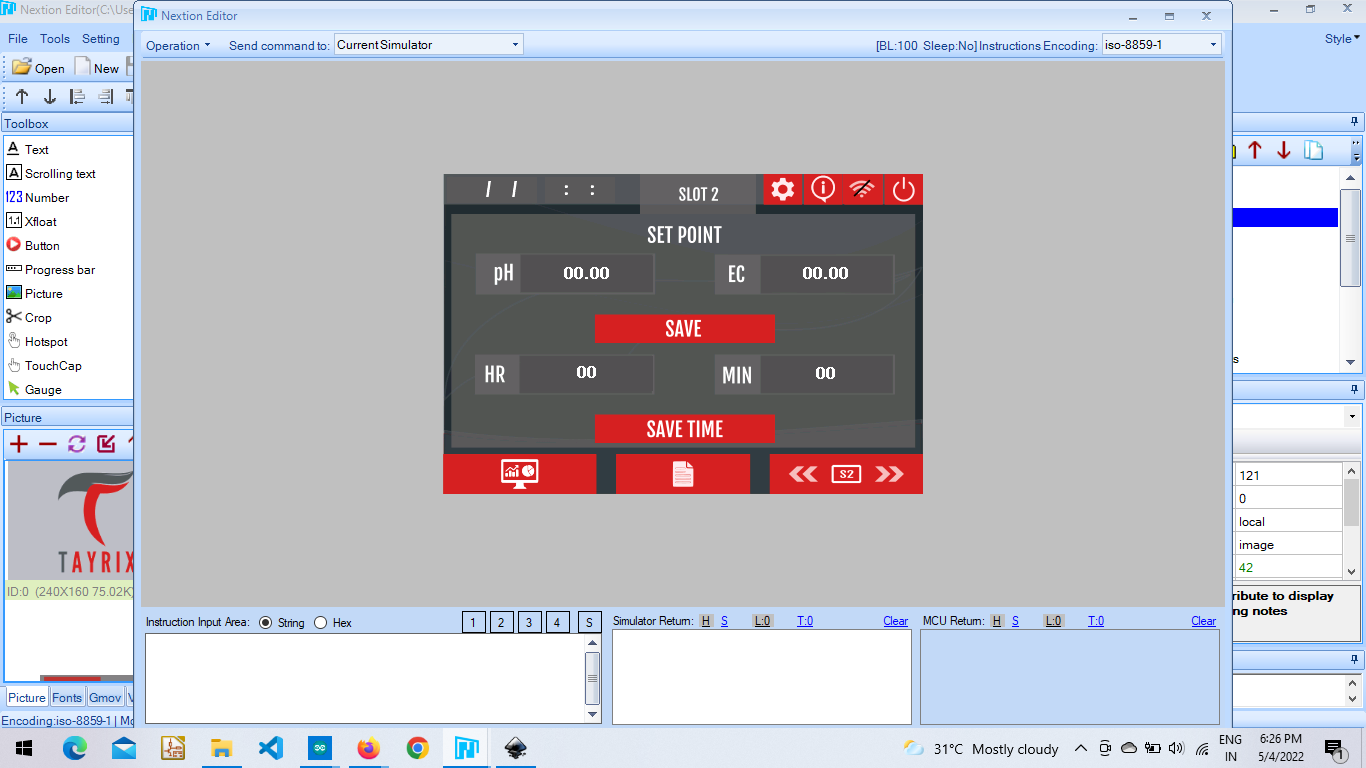
****

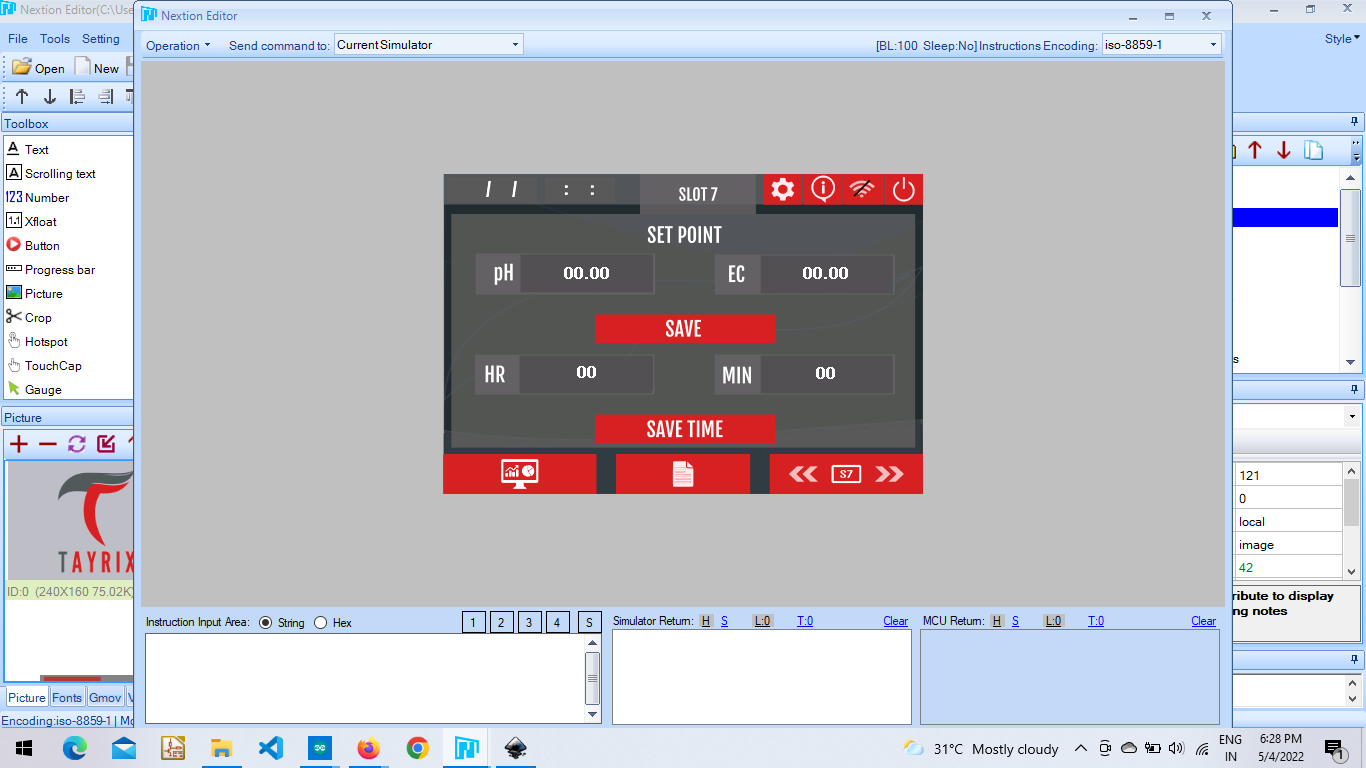
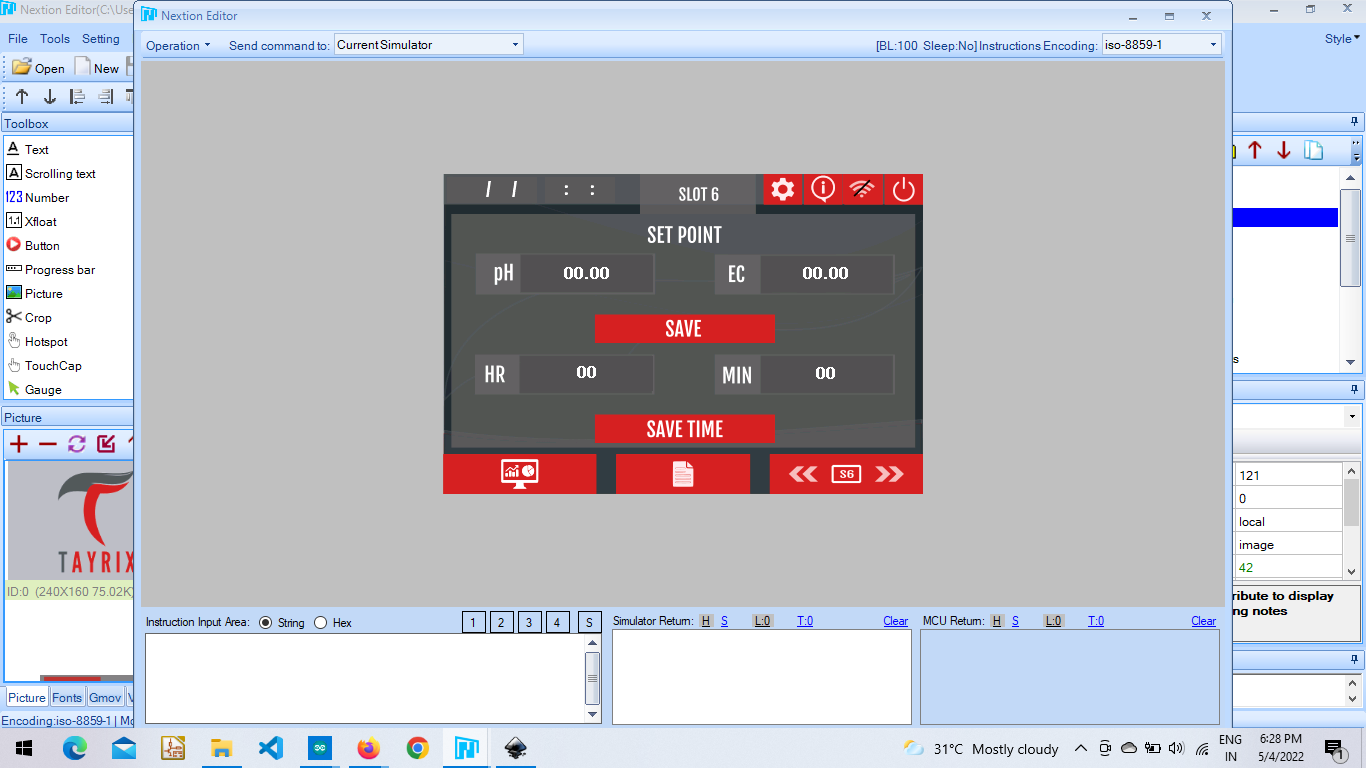
This page is used to enter the values. Ph and Ec values required for plants and the time when these values should effect. After entering the user needs to save them and go back to the schedule page to ON/OFF the Mode.

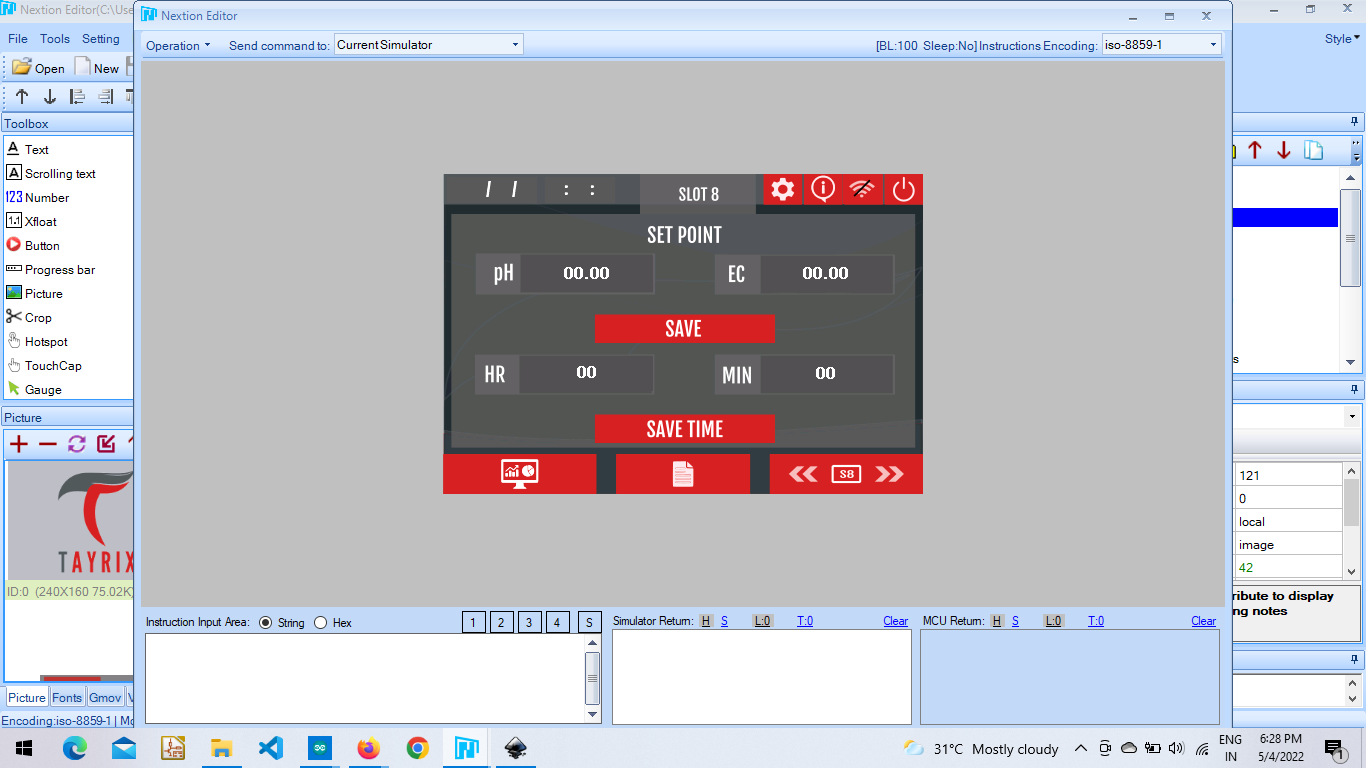
There is a code return for each save button.

****

Similarly the same is for remaining 7-slots.

****

****

****

**func\_name(){**

**func\_name(){**

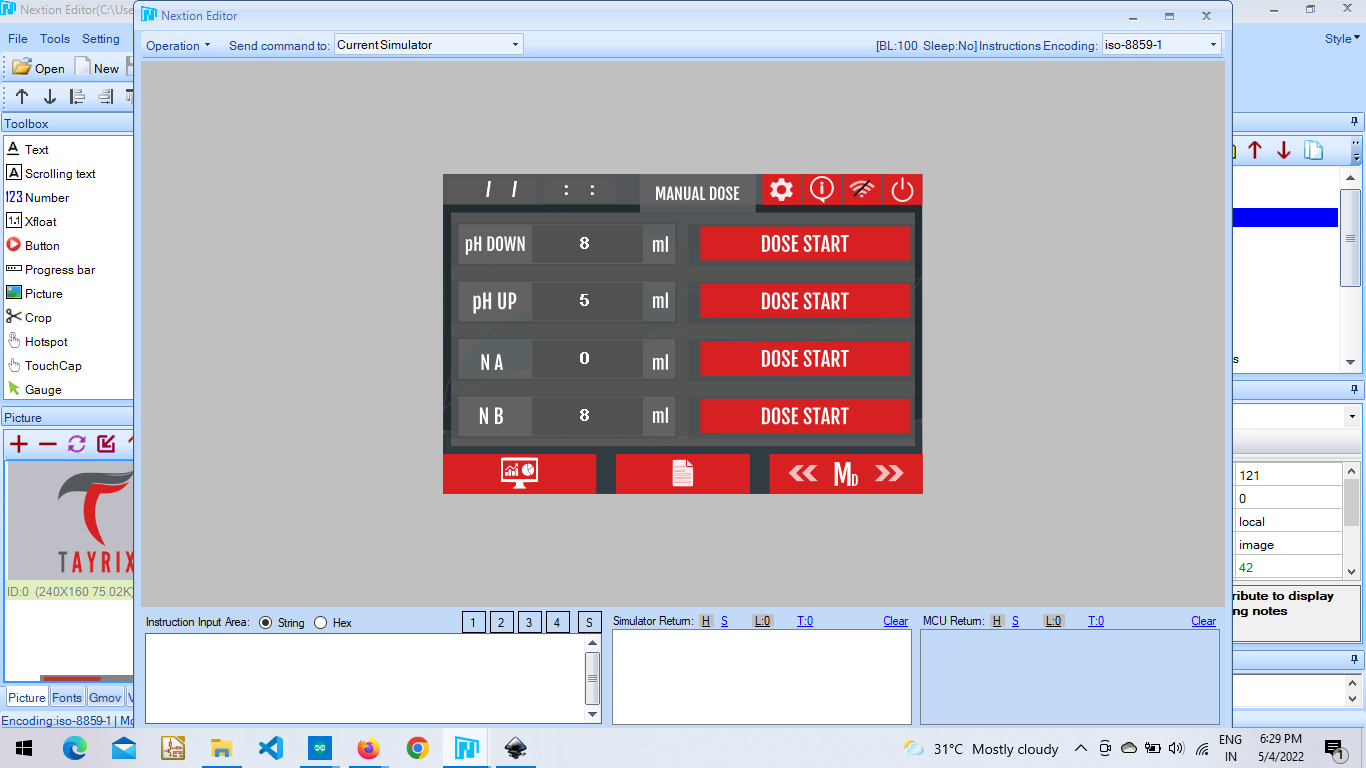
**………..**

**………..**

**………..**

**}**

**Manual Mode :** In this mode users can decide how much quantity of ph, ec, nutrients to add in millilitres. There is a code return for the “Doser Start” button to perform the need.

**.**

**func\_name(){**

**func\_name(){**

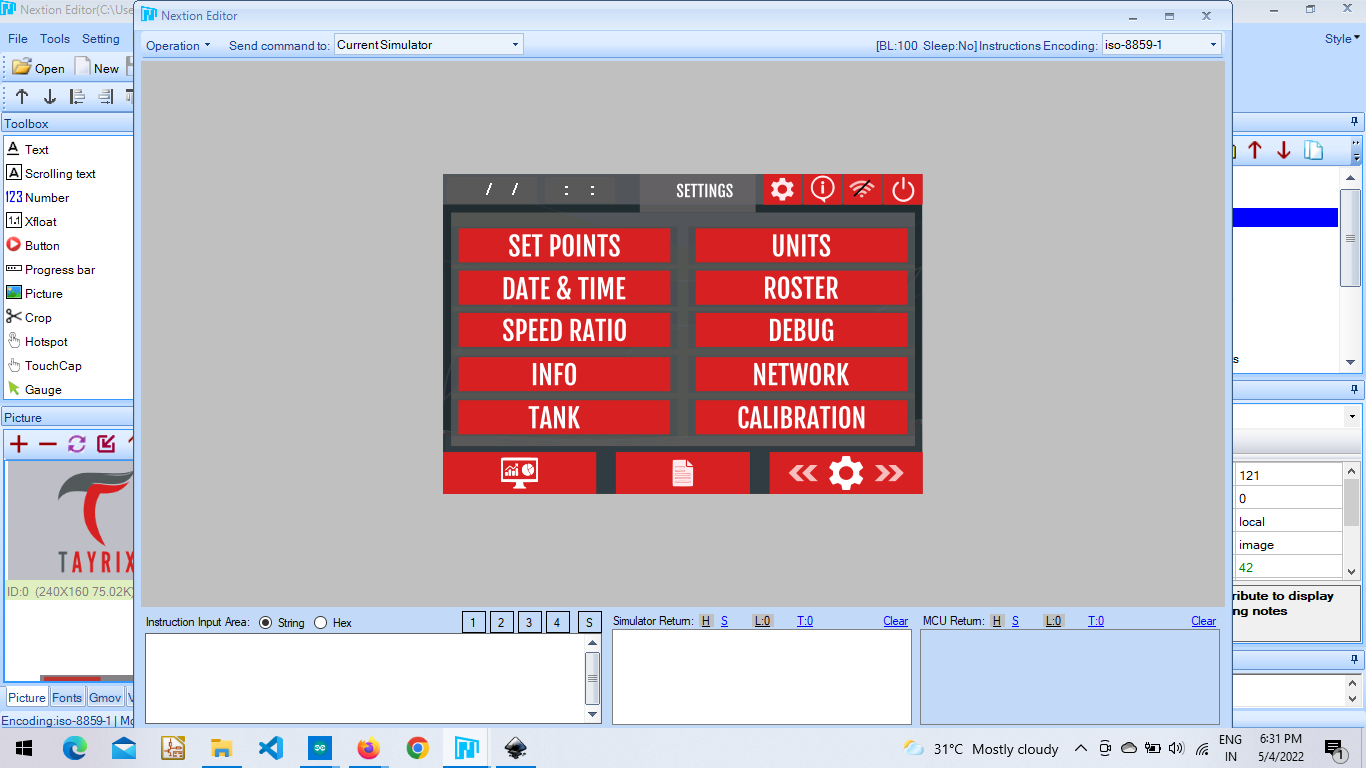
**………..**

**………..**

**………..**

**}**

**Settings Page :** This page consists of the different options.

****

**Set Points :** In this page the hysteresis values of the ph and ec sensor is given depending on the current ph and ec value and our requirement, in the auto mode page. There is a code return for the save button to send values to the arduino.

****

**func\_name(){**

**func\_name(){**

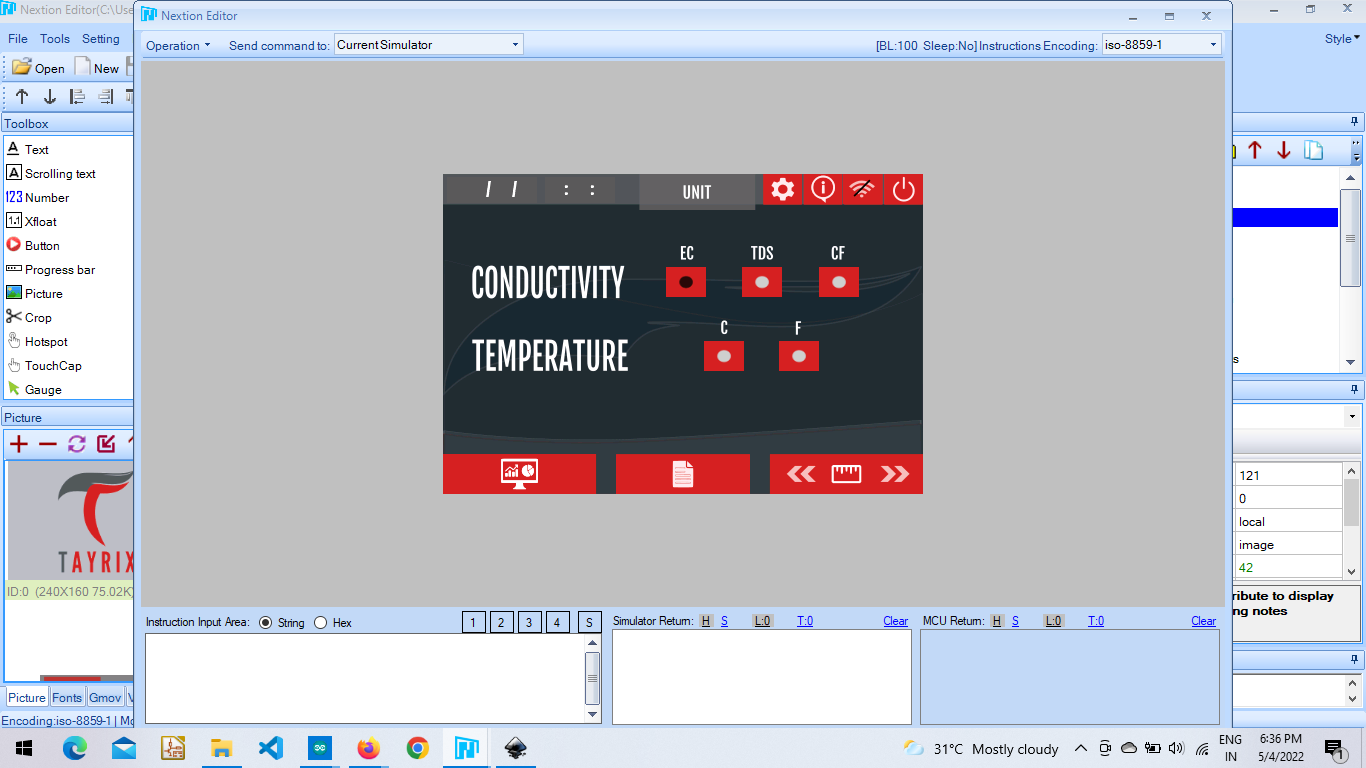
**………..**

**………..**

**………..**

**}**

**Units :** From this page users can select what type info wants to be displayed on the Home page. For each button there is code return.

****

**func\_name(){**

**func\_name(){**

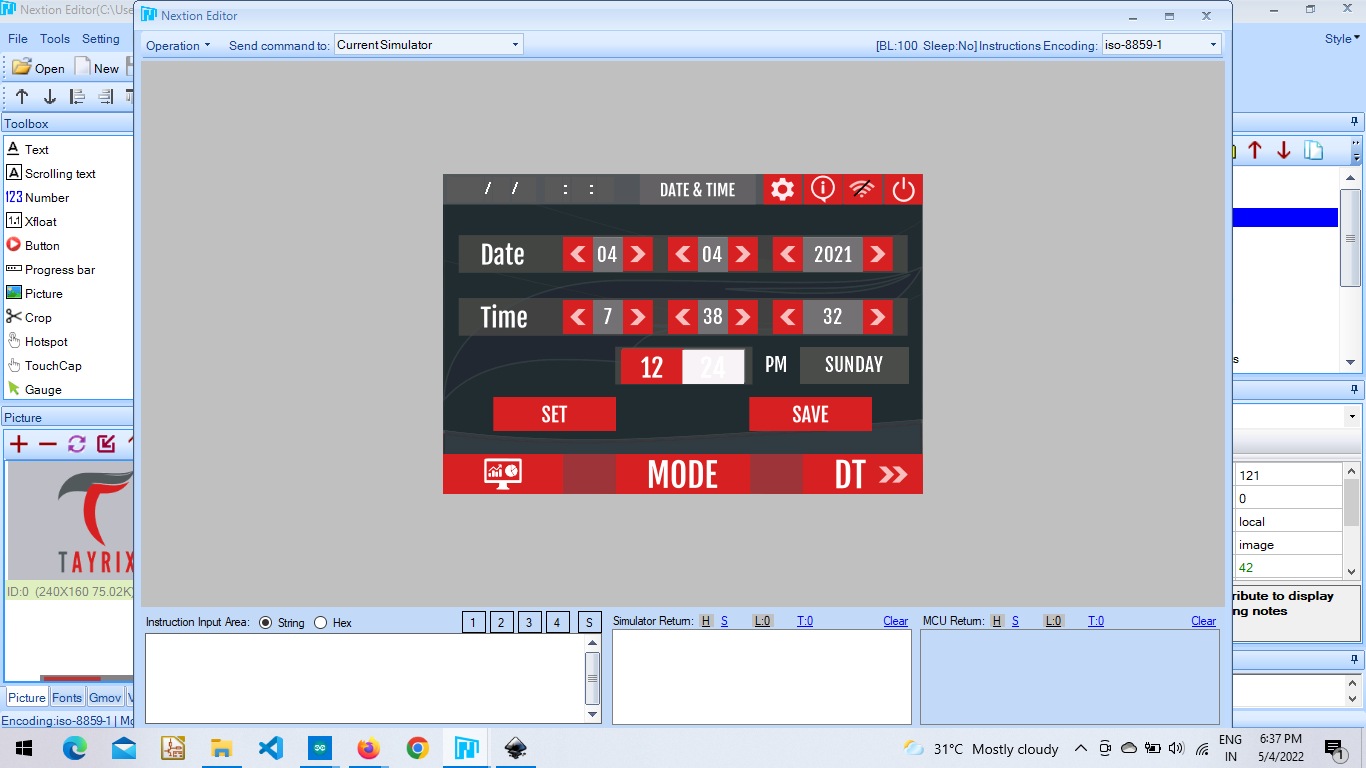
**………..**

**………..**

**………..**

**}**

**Date & Time :** This page is used to set the time on nextion device. Here we need to first set the current time and then save it. When the save button is clicked the values are sent to the arduino and when the set button is clicked the arduino sends back the selected time after updating its RTC. Thus both Arduino Rtc and Nextion Rtc will be running at the same time.

****

**func\_name(){**

**func\_name(){**

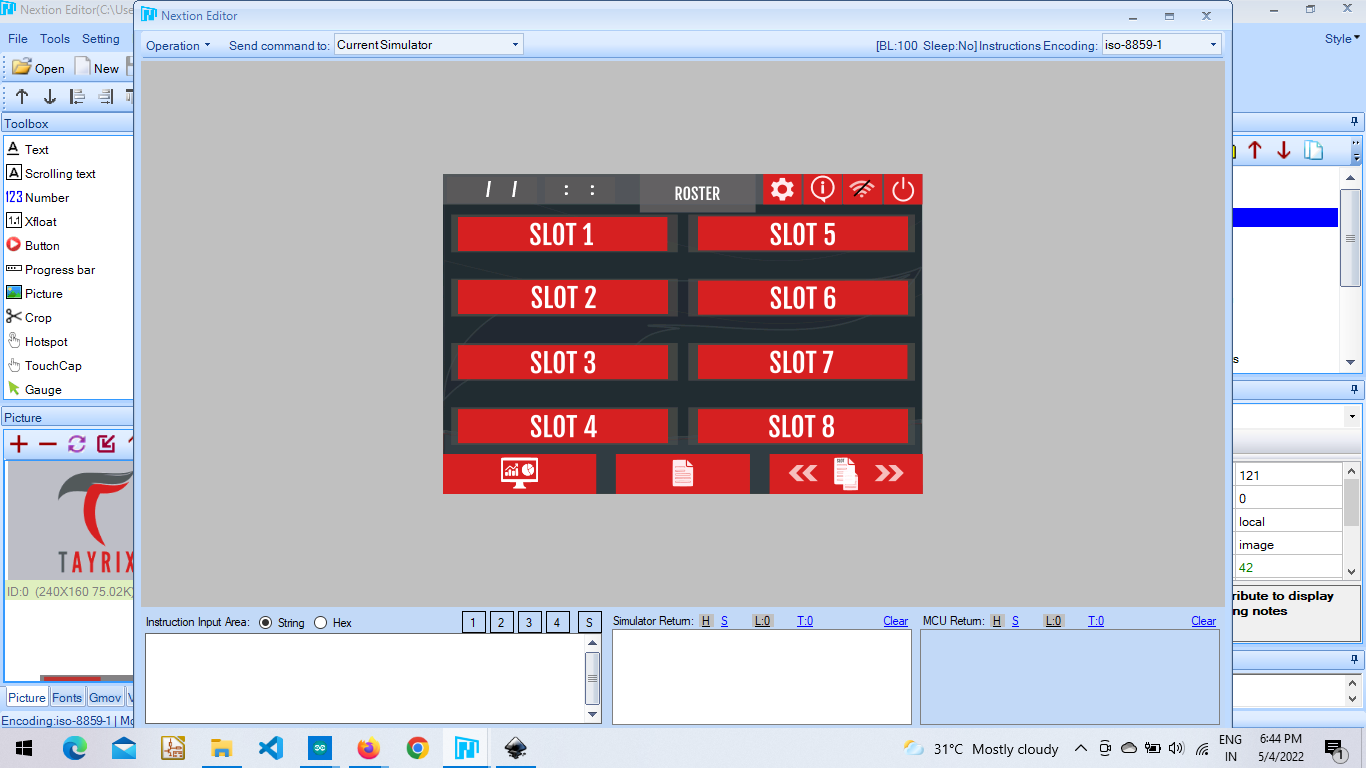
**………..**

**………..**

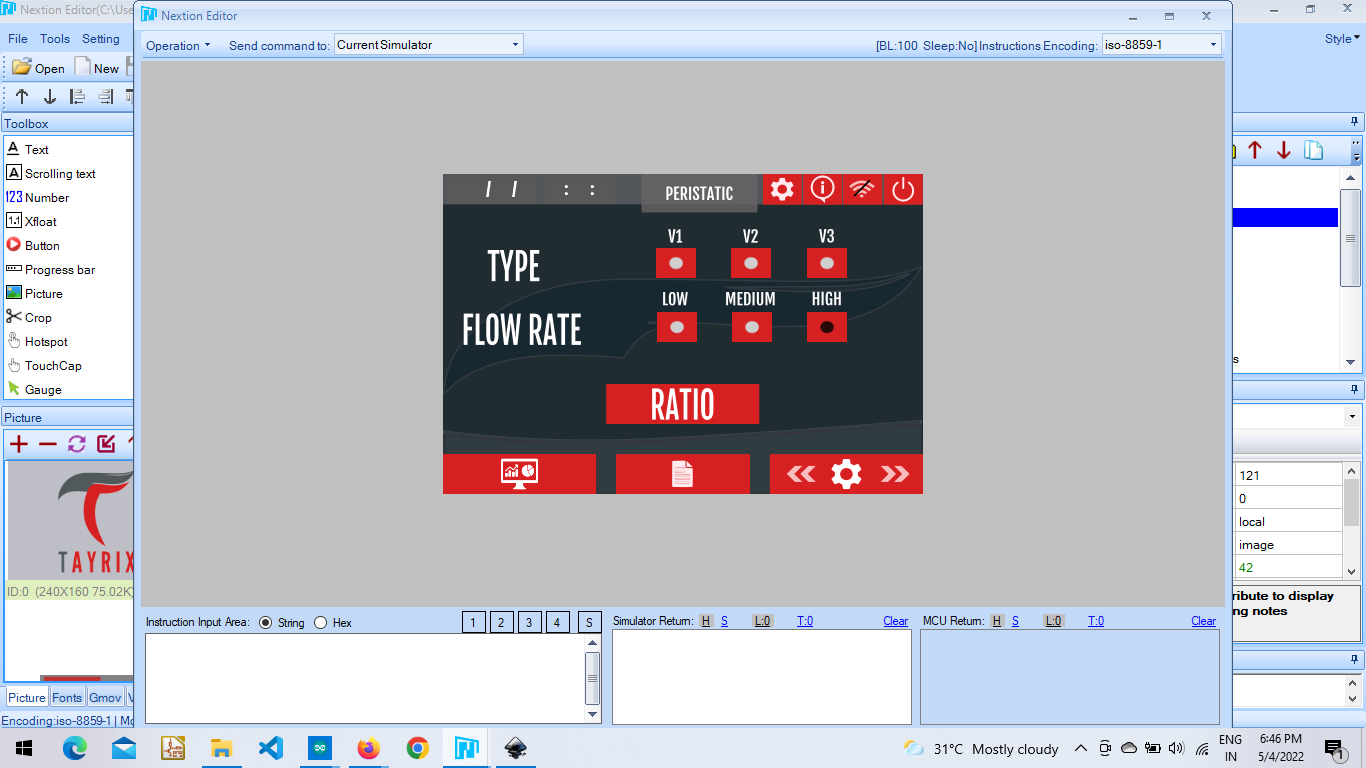
**………..**

**}**

**Roster :** This page is similar to 8-slots of schedule mode and performs the same actions.

****

**Speed Ratio :** This page is named peristaltic and it shows past and current versions of the dosing controller. And users can select the speed at which the nutrients to be added according to requirement. The code is written for each button and the ratio button navigates to another page.

****

**func\_name(){**

**func\_name(){**

**………..**

**………..**

**………..**

**}**

**}**

**func\_name(){**

**func\_name(){**

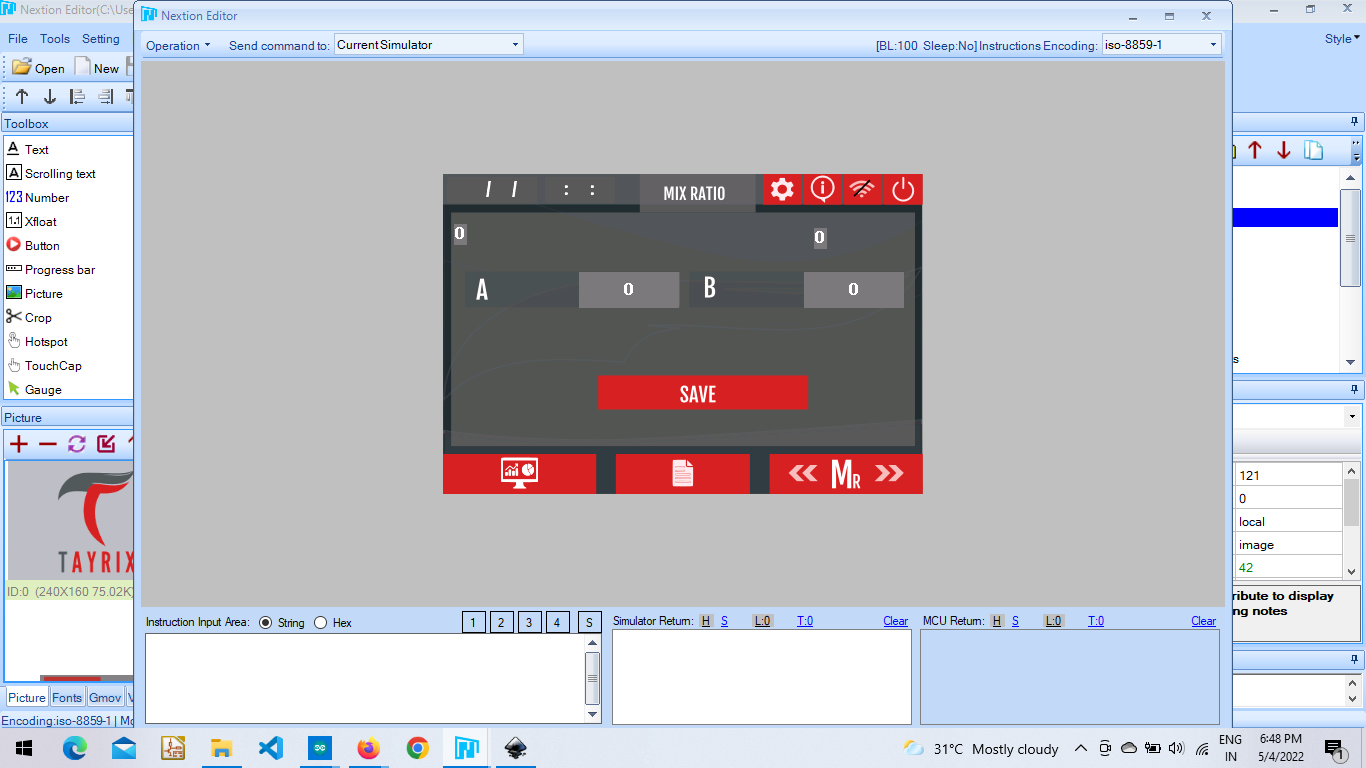
**………..**

**………..**

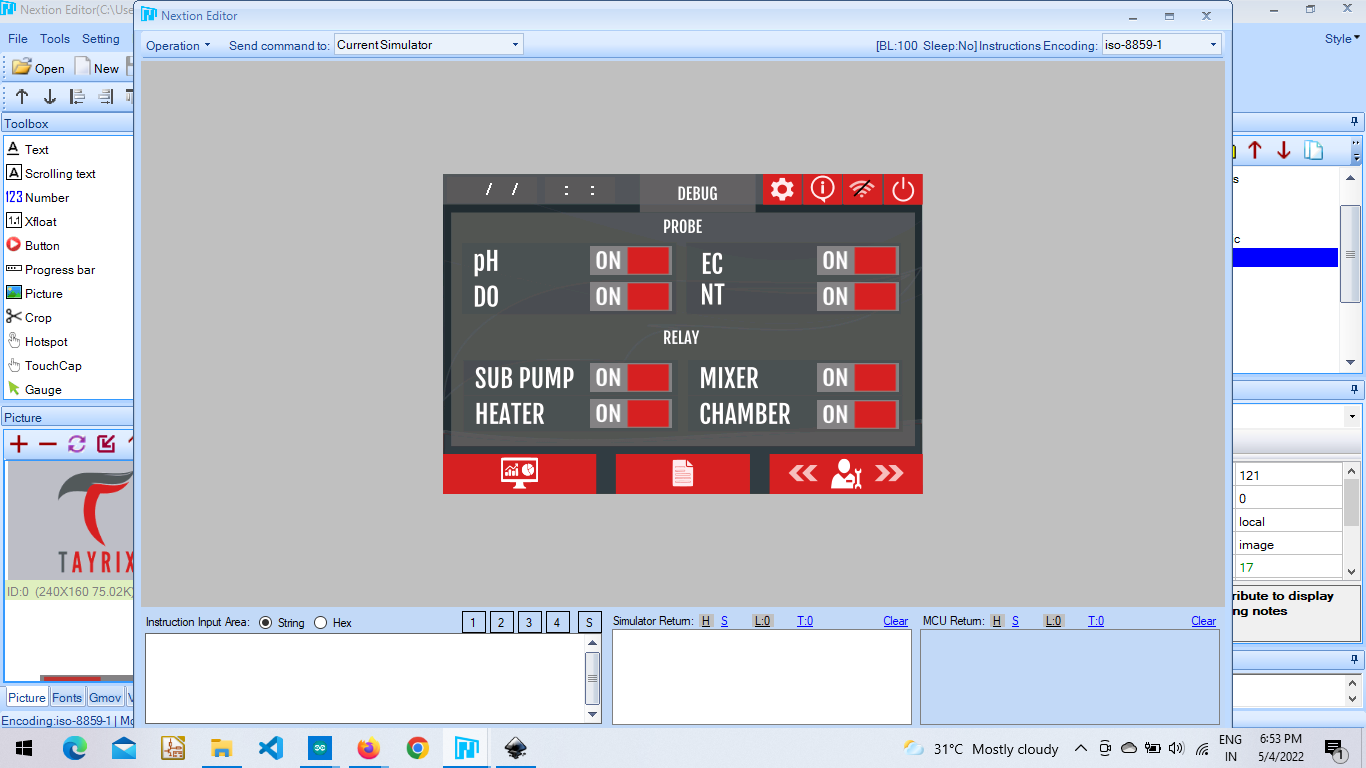
**………..**

**}**

The mix ratio page is how much ratio the user wants to use. There is code return for save button to send values to arduino

****

**Debug :** The ph, ec, do & nt toggle buttons consist of code which reflect the homepage whether to display or not the values coming from arduino. And the remaining submersible pump, mixer, heater and chamber are connected to the relays of the dosing controller. These toggle buttons reflect the relays on/off state respectively.

****

**func\_name(){**

**func\_name(){**

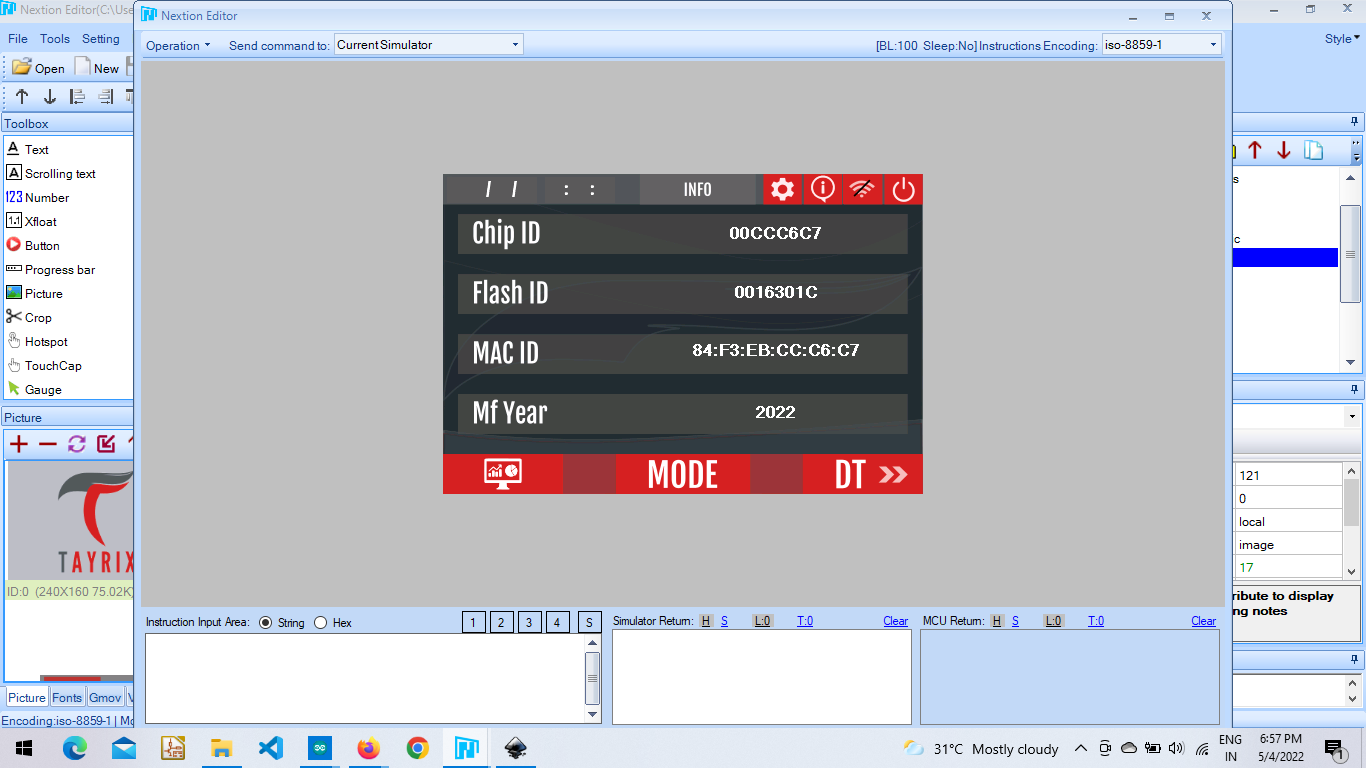
**………..**

**………..**

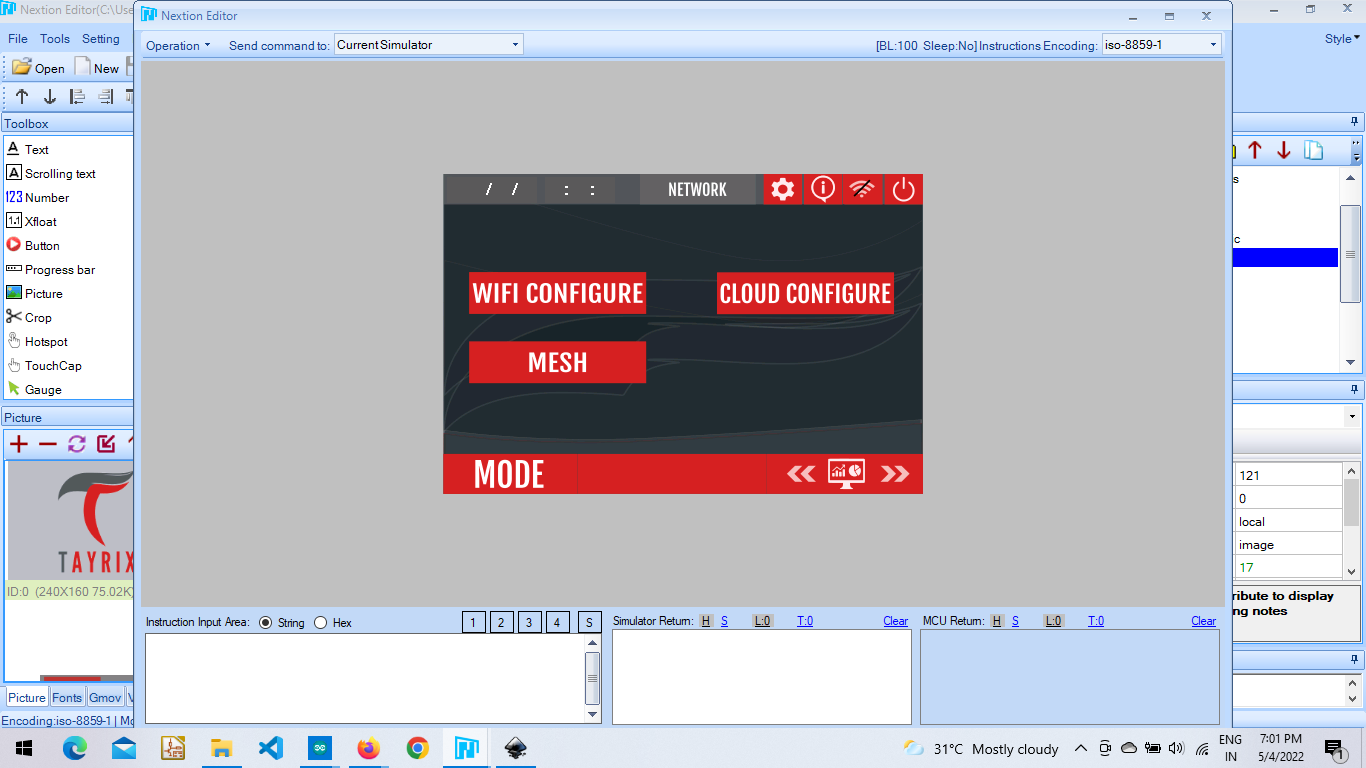
**………..**

**}**

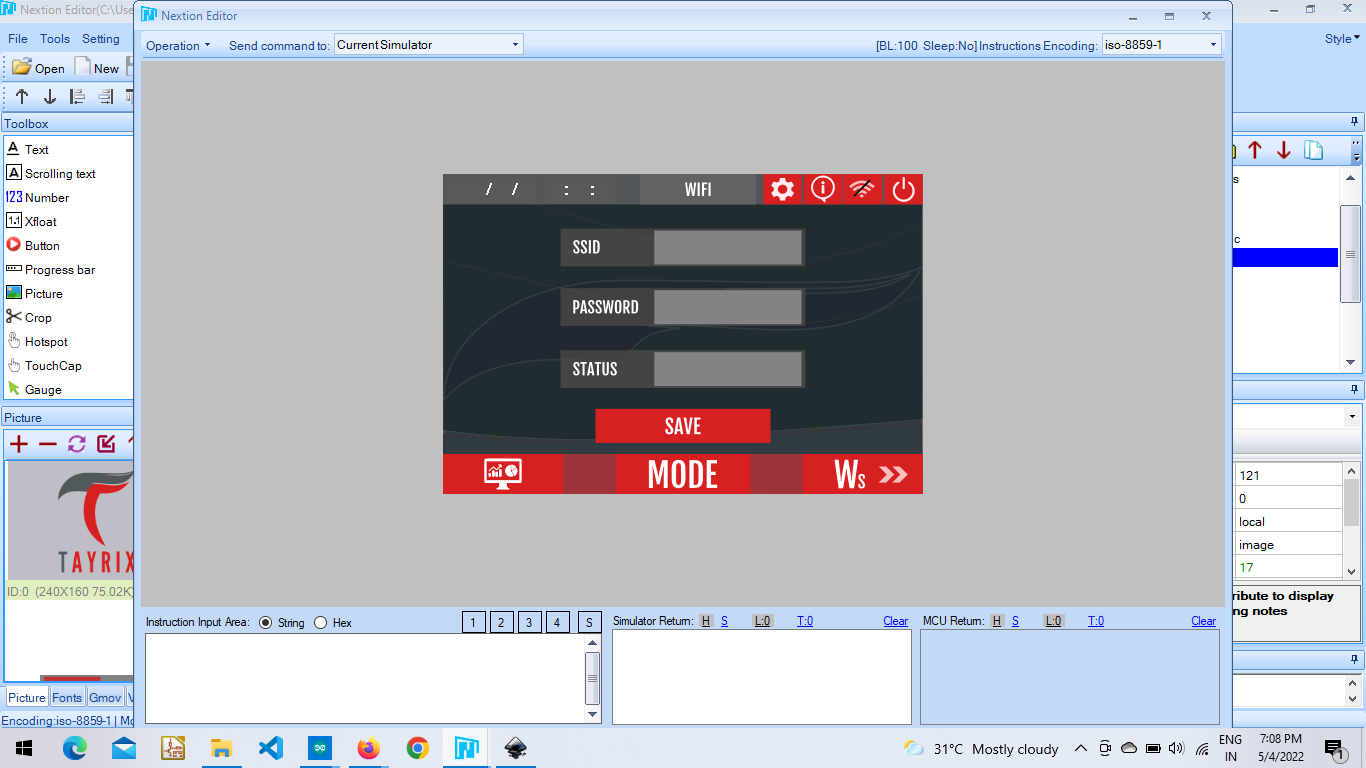
**Info :** The chip id and flash id is shown from arduino mega after receiving from esp 8266.

****

**Network :** This page consists of wifi configuration, cloud and mess configuration.

****

**Wifi Configuration :** User needs to enter the SSID and Password in order to connect to wifi so that the esp 8266 can start working to send data collected from sensors and relays to the cloud. The box “status” takes the signal from esp 8266 whether the wifi is connected or/not and sends the signal to arduino mega from where it is shown here.

****

**func\_name(){**

**func\_name(){**

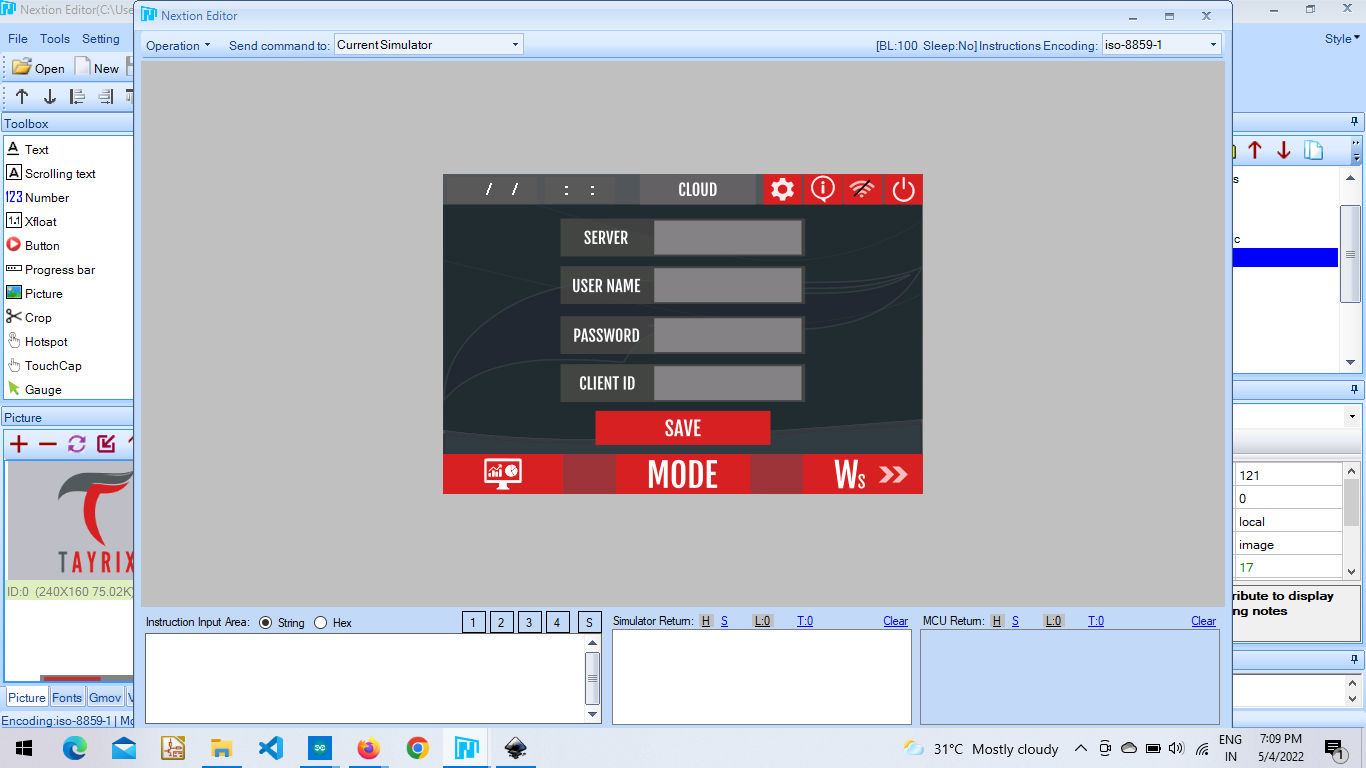
**………..**

**………..**

**………..**

**}**

**Cloud Configuration :** This page is to enter credentials to connect to the cloud.

****

**func\_name(){**

**func\_name(){**

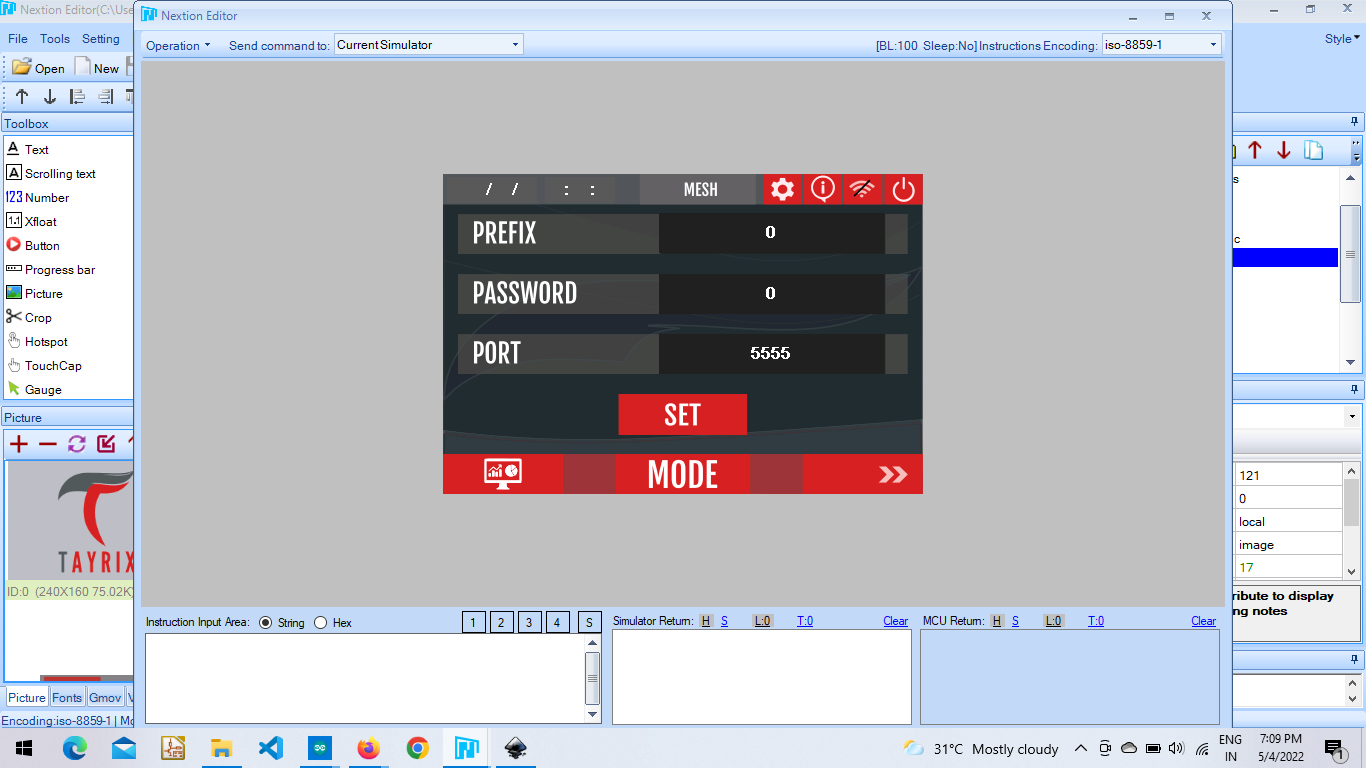
**………..**

**………..**

**………..**

**}**

**Mesh Configuration :** This page is designed to connect the Master node.

****

**func\_name(){**

**func\_name(){**

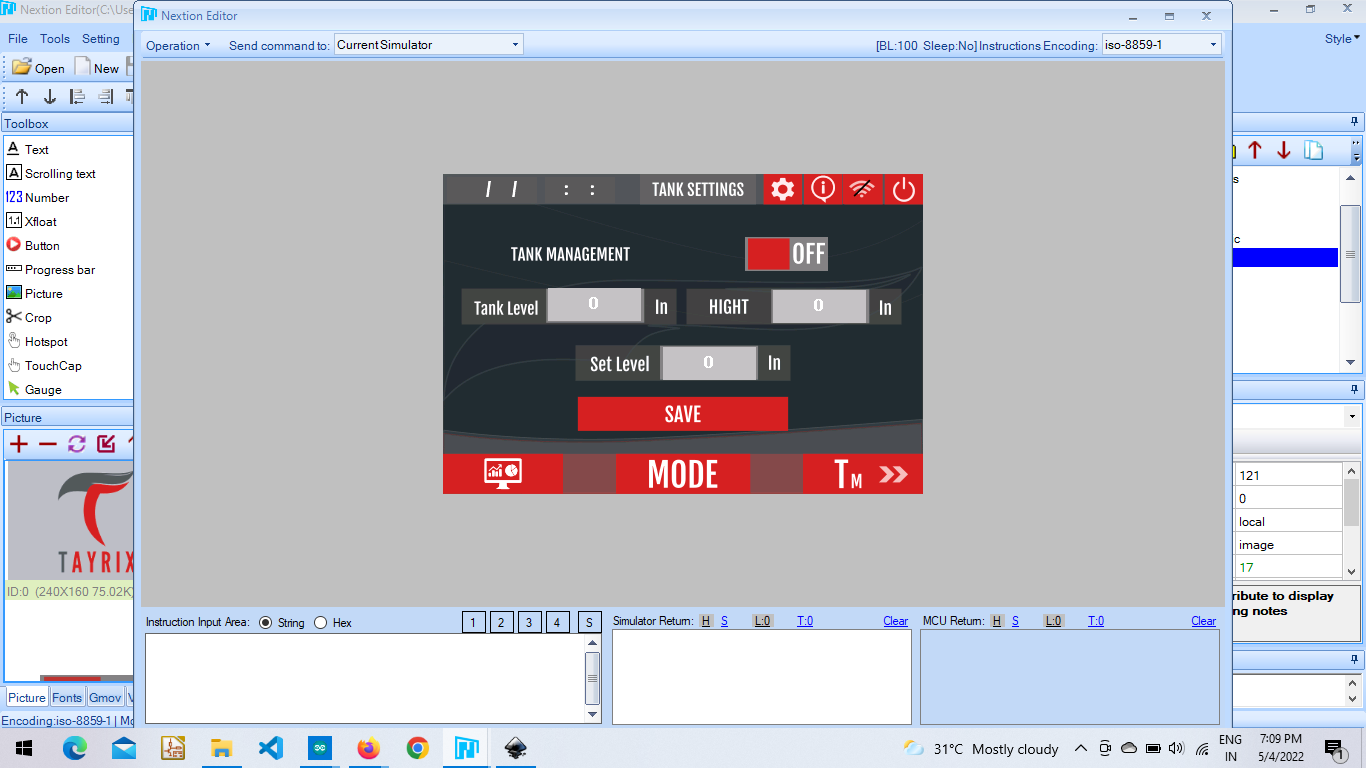
**………..**

**………..**

**………..**

**}**

**Tank Settings :** This page is used to control the water level inside the tank depending on the requirement. The user needs to give how much is the height of the tank and the requirement needed i.e. level of tank in inches in “Set Level Box”. The calculation after receiving values from the sensor is done within arduino mega and displayed in the “Tank Level Box”. Users need to save the values of height and set level then select the ON/OFF toggle button, to run the tank management system. If the tank level is less than the set level then the RO Pump is on flipping up the tank depending on requirement.

****

**func\_name(){**

**func\_name(){**

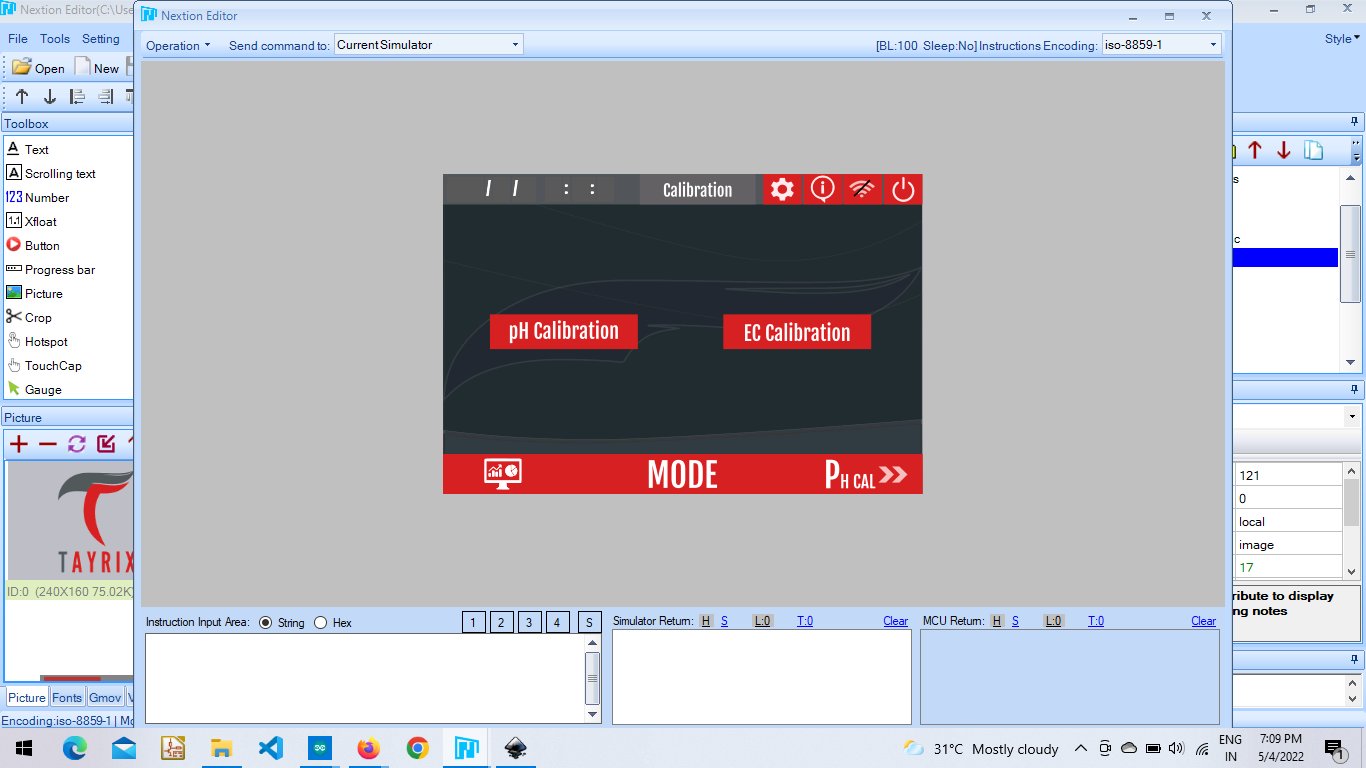
**………..**

**………..**

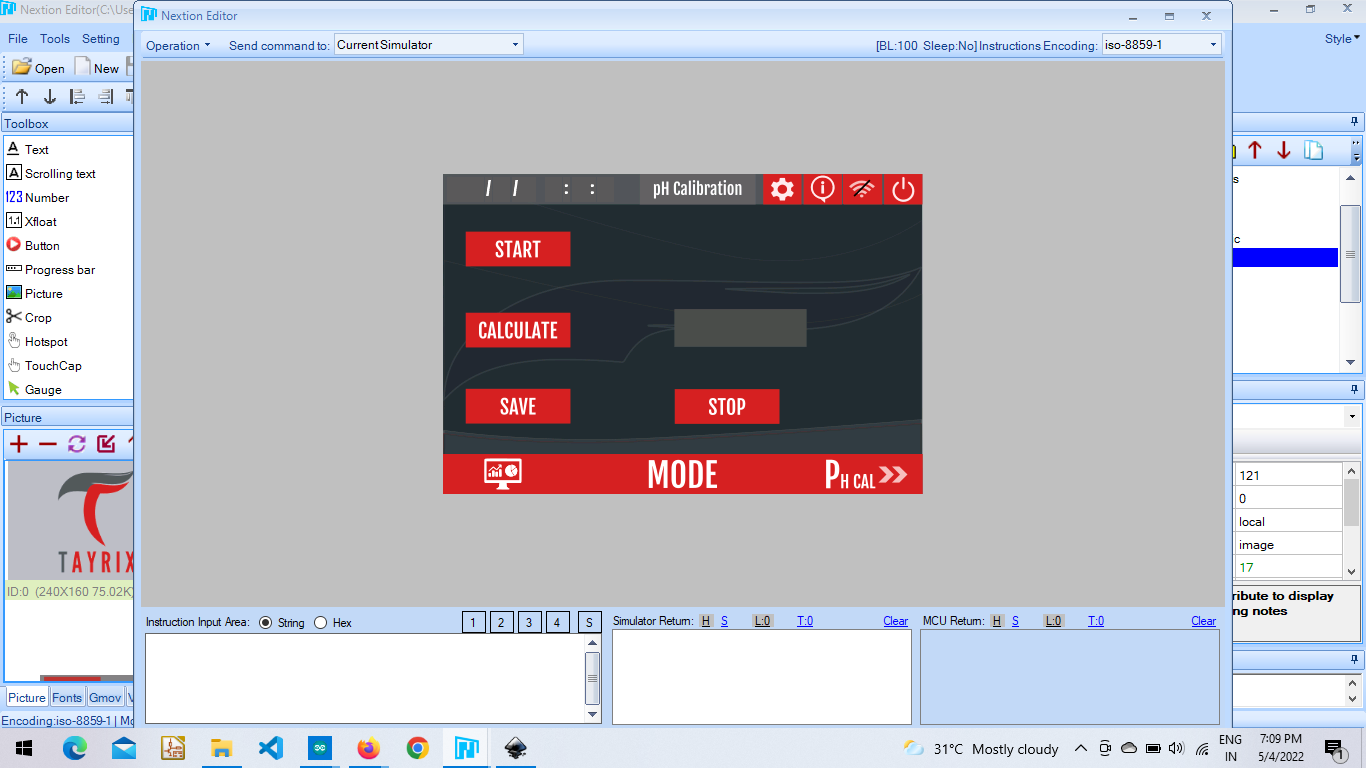
**………..**

**}**

**Calibration :** The pH and Ec sensors need calibration once a while to give accurate values when dipping in running water from a tank passing through plants. Here both sensors have calibration parts separate for each.

****

**pH Calibration :** Coming to ph calibration part, the user needs to click on start to start calibration mode, then click on calculate to start calculation, after the sensor starts showing accurate values click on save button and stop the calibration mode.

****

**func\_name(){**

**func\_name(){**

**………..**

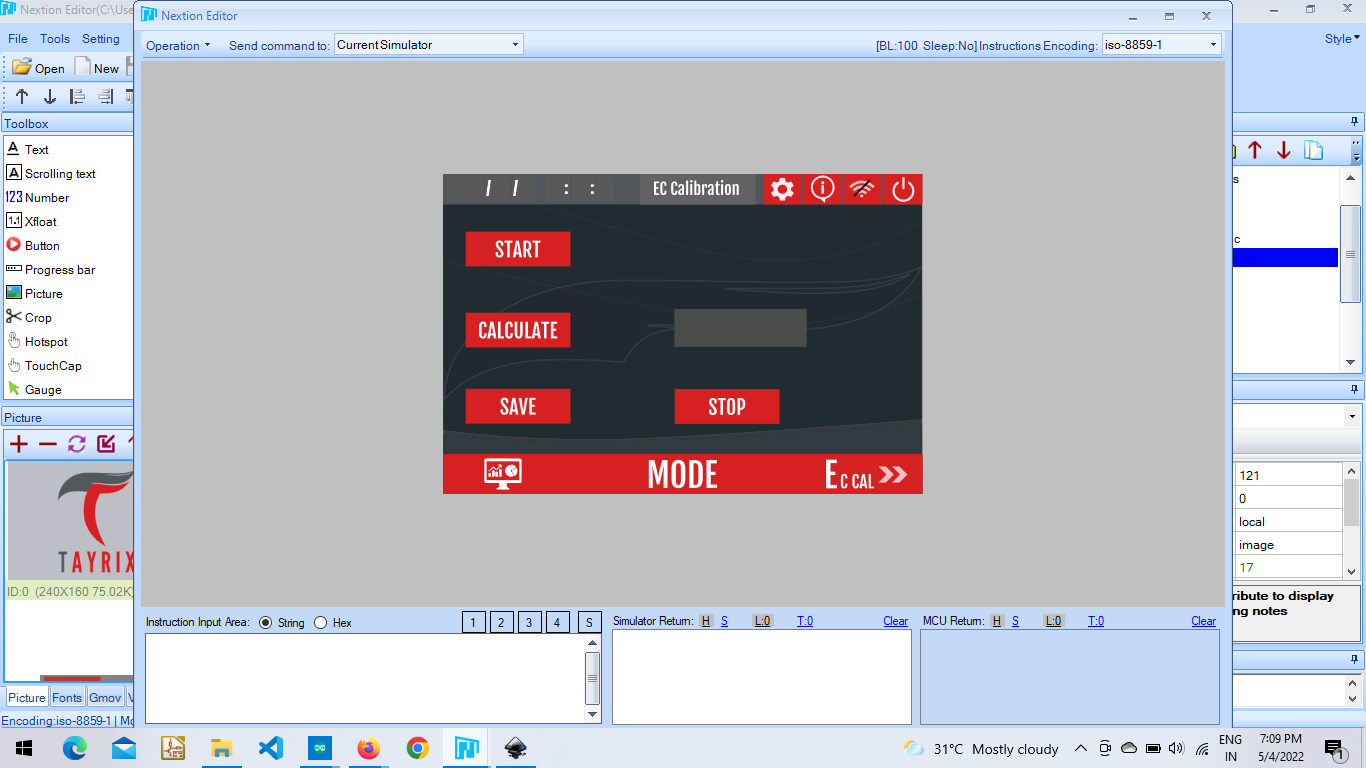
**………..**

**………..**

**}**

**Here t16 is based on a task scheduler in which the calibration() function is running.**

**Ec Calibration :** Coming to Ec calibration part, the user needs to click on start to start calibration mode, then click on calculate to start calculation, after the sensor starts showing accurate values click on save button and stop the calibration mode.

****

**func\_name(){**

**func\_name(){**

**………..**

**………..**

**………..**

**}**

**Credits :** This project has been undertaken by Mr. Raj Kumar. Ardunio Mega and Esp 8266 coding was done by Him Only. Under his guidance I have worked on Nextion to Arduino and vice versa communication.

**Contact Information :**

Feel free to contact the address below for further help.

E-mail Id **:** [**01zeptohas@gmail.com**](mailto:01zeptohas@gmail.com)