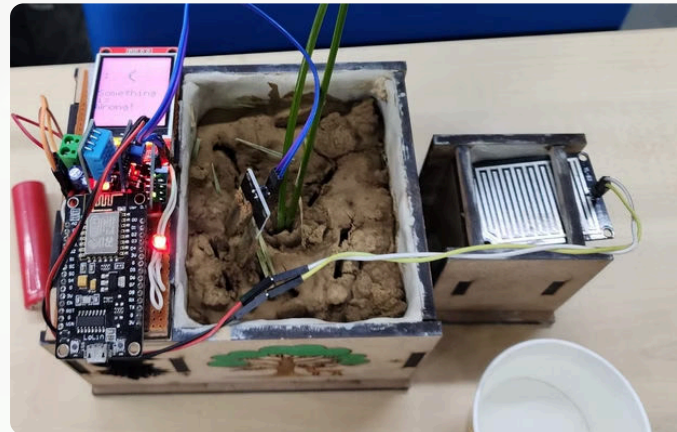
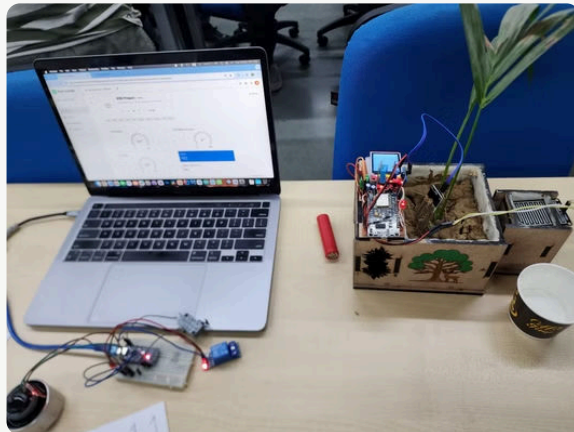


Smart IoT Botany STEM Kit



ESD (ECE211) Course Project.

Abhishek Jha (2022023)

Group11

Team Name: IronMan.

Project Overview

Project Overview:

The Smart IoT Botany STEM Kit is an innovative educational tool designed to engage and inspire teenage kids aged 10-13 in the captivating world of botany and STEM (Science, Technology, Engineering, and Mathematics) learning. This comprehensive kit provides hands-on, interactive experiences that foster a passion for science and the natural environment, empowering the next generation of young botanists and scientists.

Introduction:

In today's digital age, fostering an interest in science and nature among young learners is more important than ever. The Smart IoT Botany STEM Kit represents a unique fusion of technology and botany, offering an immersive learning experience that combines hands-on exploration with cutting-edge IoT (Internet of Things) technology.

Designed with teenage kids aged 10-13 in mind, this kit introduces students to the fundamentals of plant care and horticulture while integrating real-time monitoring and control capabilities through IoT tech. By leveraging sensors, microcontrollers, and a user-friendly app interface, students can actively monitor and nurture their plants, gaining valuable insights into plant growth and environmental conditions.

Through gamification features like leaderboards and streaks, the Smart IoT Botany STEM Kit transforms plant care into an engaging and rewarding experience, encouraging students to develop critical thinking, problem-solving, and digital literacy skills. Ultimately, this innovative STEM solution empowers students to explore the fascinating world of botany while cultivating a sense of responsibility towards the environment.



Project Objectives

The primary objective of the Smart IoT Botany STEM Kit is to provide a comprehensive and engaging platform for teenage kids aged 10-13 to explore the captivating realms of botany, STEM (Science, Technology, Engineering, and Mathematics) learning, and environmental stewardship. This multifaceted project aims to achieve the following goals:

1. **Introduce Botanical Concepts:** Foster a deeper understanding and appreciation for botany and plant science among teenage learners through hands-on exploration and experimentation.
2. **Utilize IoT Technology:** Harness the power of IoT (Internet of Things) technology to enable real-time monitoring and remote control of plant growth parameters, enhancing the learning experience and promoting technological literacy.
3. **Promote Environmental Awareness:** Cultivate a sense of responsibility and environmental stewardship by emphasizing the importance of sustainable plant care practices and ecosystem conservation.
4. **Engage and Motivate Users:** Gamify the plant care experience by incorporating features such as leaderboards, streaks, and rewards, motivating users to actively participate in nurturing their plants and fostering a sense of achievement.
5. **Target Audience:** Tailor the Smart IoT Botany STEM Kit to meet the needs and interests of teenage kids aged 10-13, a demographic characterized by curiosity, eagerness to learn, and responsiveness to hands-on, interactive educational experiences.
6. **Remote Software Control through Blynk:** Enable users to remotely monitor and control their plants using the Blynk app, providing a seamless and intuitive interface for adjusting settings, receiving alerts, and accessing real-time data from anywhere.
7. **Plug and Play Functionality:** Design the Smart IoT Botany STEM Kit with plug-and-play functionality for ease of setup and operation, ensuring that users can quickly assemble the kit and start exploring without the need for extensive technical knowledge or experience.

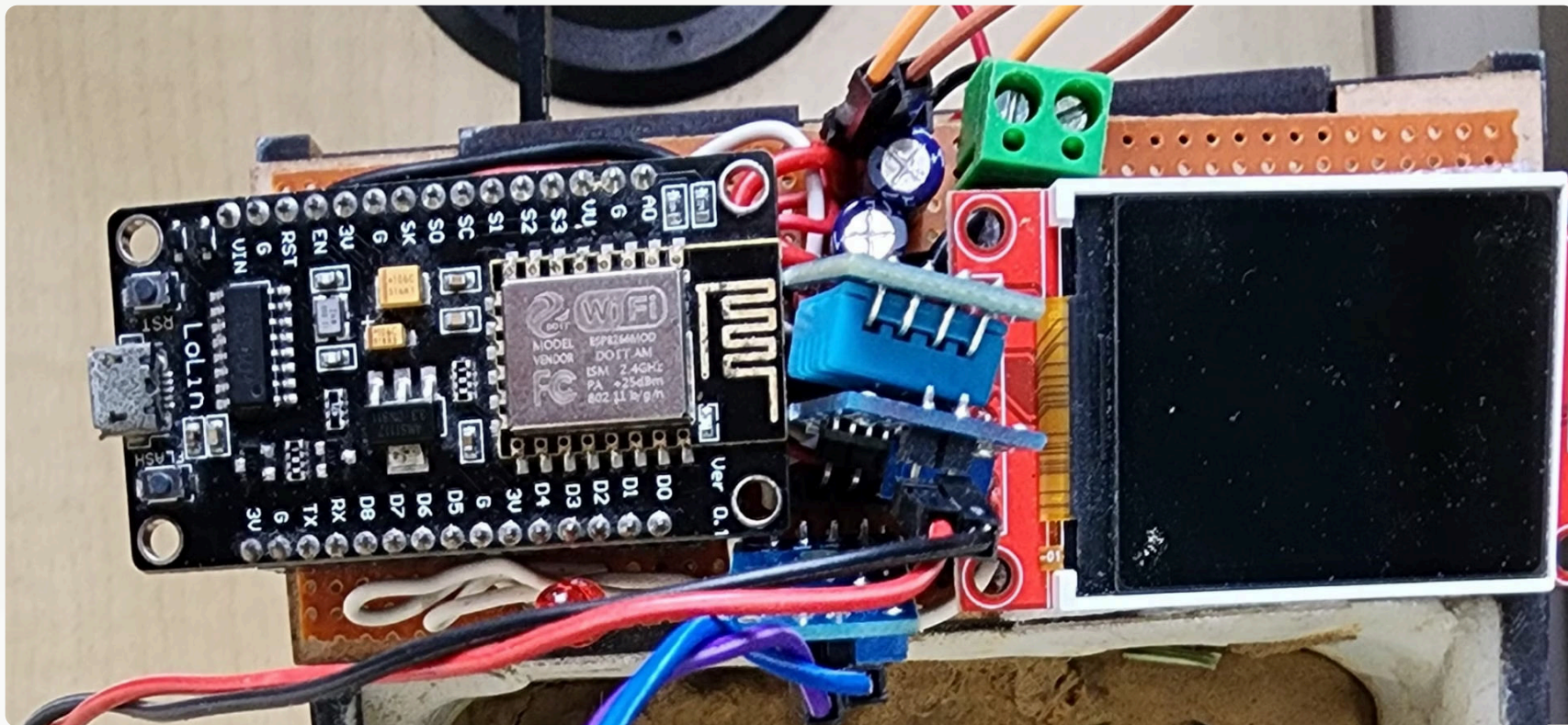
Project Components

Hardware Components	- Microcontroller Unit (e.g., Arduino or NodeMCU) - Sensors: Soil Moisture, Water Level, Temperature & Humidity - LCD Display Screen - Enclosure (MDF board, acrylic panels) - Power Supply (batteries or power adapter) - Connecting Wires and Breadboard
Software Components	- Arduino IDE for programming the microcontroller - Blynk app for remote monitoring and control - Libraries for sensor interfacing and data processing
Educational Resources	- Curriculum materials (lesson plans, worksheets) - Online resources (videos, articles) for further learning - Gamification elements (leaderboards, achievements)
Miscellaneous Components	- Waterproofing material (MSEAL clay) for electronics - Planting pot or tray for the plant - Mounting hardware (screws, nuts) for assembly
Documentation and Instructions	- Assembly instructions - User manual for setup, operation, and troubleshooting - Educational guides on botany, environmental science, STEM
Packaging and Branding	- Packaging materials (box, labels) for safe transport - Branding elements (logo, colors) for visual identity

NodeMCU Microcontroller

The NodeMCU ESP8266 is a highly capable and compact microcontroller board that boasts built-in Wi-Fi connectivity. Powered by the powerful ESP8266 chip, it offers a 32-bit processor, ample GPIO pins, and generous flash memory for running diverse IoT applications.

With support for the Arduino IDE and Lua scripting, NodeMCU makes it easy to build innovative connected solutions, from home automation to environmental monitoring systems. Its affordable price point and vibrant community support make it a popular choice for hobbyists and professionals alike.



Node MCU PinOut

NodeMCU Development Board

Pinout Configuration

Pin Category	Name	Description
Power	Micro-USB, 3.3V, GND, Vin	<p>Micro-USB: NodeMCU can be powered through the USB port</p> <p>3.3V: Regulated 3.3V can be supplied to this pin to power the board</p> <p>GND: Ground pins</p> <p>Vin: External Power Supply</p>
Control Pins	EN, RST	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

Methodologies used-:

Design Thinking Process

- Empathize: Understanding the needs, motivations, and pain points of the target audience (teenage kids aged 10-13).
- Define: Defining the problem and identifying opportunities for innovation.
- Ideate: Generating creative solutions to address the defined problem statement.
- Prototype: Developing tangible prototypes to test and refine ideas.
- Test: Testing the prototypes with real users to gather feedback and iterate on the design.
- Implement: Bringing the final design to life and preparing it for production.

Laser Cutting of MDF Boards

- Utilizing laser cutting technology to precisely cut and engrave Medium-Density Fiberboard (MDF) for the prototyping base.
- Using Computer-Aided Design (CAD) software such as AutoCAD to create precise designs and layouts for the MDF enclosure.

Electronics Methods

- Selecting appropriate electronic components such as sensors, microcontrollers, and displays for the project.
- Wiring and soldering the electronic components according to circuit diagrams and schematics.
- Programming the microcontroller unit (e.g., Arduino or NodeMCU) using the Arduino IDE or other compatible software to control sensor readings, display outputs, and communication with the IoT platform.

IoT (Internet of Things)

- Integrating IoT technology into the project to enable real-time monitoring and control of plant growth parameters.
- Using IoT platforms such as Blynk to develop a smartphone app for remote monitoring and control of the Smart IoT Botany STEM Kit.
- Establishing Wi-Fi connectivity between the microcontroller unit and the IoT platform to facilitate data communication and user interaction.

By employing these methodologies and techniques, the project team was able to design, develop, and implement the Smart IoT Botany STEM Kit, providing an engaging and educational experience for teenage kids while fostering their interest in botany and STEM learning.

Software For Cloud Networking:

Blynk (Visit Website: <https://blynk.io/>)

- Blynk is an intuitive IoT platform designed for creating IoT projects and applications.
- Users can build custom interfaces for controlling and monitoring IoT devices using a smartphone app.
- The platform offers drag-and-drop functionality for easy interface design.
- Blynk supports a variety of hardware platforms and communication protocols.
- It caters to both beginners and experienced developers, providing accessibility and advanced features.
- With Blynk, users can quickly prototype and deploy IoT solutions with minimal time and effort.



WHY BLYNK?

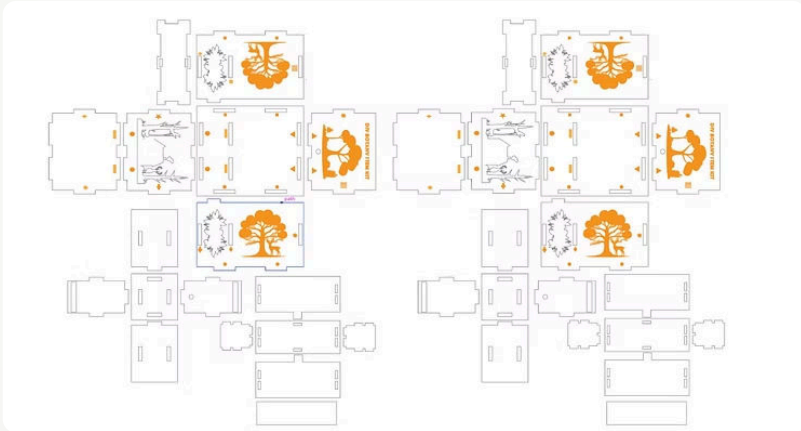
Here we are controlling LED bulb on off from remote area .

**WITH BLYNK WE CAN CONTROL OUR SYSTEM
FROM ANY PART OF THE WORLD
AS IT WORKS ON CLOUD NETWORKING BASIS.**

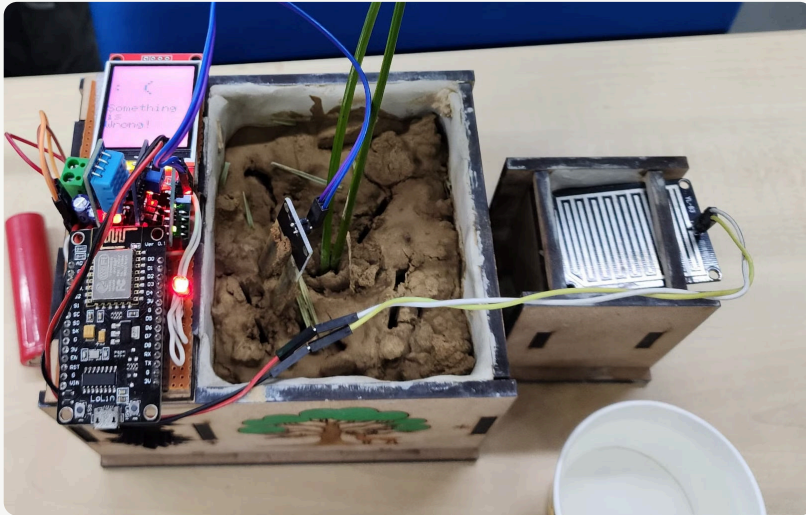
Software For Layout Of Designing Box
(AutoCad)[https://www.googleadservices.com/pagead/aclk?
sa=L&ai=DChcSEwim7I6i_I-
GAxXUWA8CHYgHAMsYABAAAgJo
Yg&ase=2&gclid=CjwKCAjwupGyBh
BBEiwAoUcqaMR-5zFVebV-
Cq08jqEPbdfB6EPpOPDP6IWSsvm
6H29Bl7KBT5ddBoCuVoQAvD_BwE
&ohost=www.google.com&cid=CAE
SVeD21xWc9Ldp4EcsHIRL_IF3eCp
7jjqX3qdYx0JTfe2jxmTsXZqqgey4Jt
On5a18mk367josUJWSAtsxPVaQ8tr
XWbtyTjK9nxFDG7BdTeG4efoKTM
w&sig=AOD64_3kY4i-
Z99qssdALNUSkkOLdAzXCA&q&nis
=4&adurl&ved=2ahUKEwjZkomi_I-
GAxV8lFYBHShjBcoQoQx6BAgGEA
E](https://www.googleadservices.com/pagead/aclk?sa=L&ai=DChcSEwim7I6i_I-GAxXUWA8CHYgHAMsYABAAAgJoYg&ase=2&gclid=CjwKCAjwupGyBhBBEiwAoUcqaMR-5zFVebV-Cq08jqEPbdfB6EPpOPDP6IWSsvm6H29Bl7KBT5ddBoCuVoQAvD_BwE&ohost=www.google.com&cid=CAESVeD21xWc9Ldp4EcsHIRL_IF3eCp7jjqX3qdYx0JTfe2jxmTsXZqqgey4JtOn5a18mk367josUJWSAtsxPVaQ8trXWbtyTjK9nxFDG7BdTeG4efoKTMw&sig=AOD64_3kY4i-Z99qssdALNUSkkOLdAzXCA&q&nis=4&adurl&ved=2ahUKEwjZkomi_I-GAxV8lFYBHShjBcoQoQx6BAgGEAE)

AutoCAD is a widely-used computer-aided design (CAD) software that allows users to create precise 2D and 3D drawings and models. It offers a comprehensive set of tools for drafting, modeling, and annotation, making it suitable for a wide range of design tasks, including designing MDF boxes.

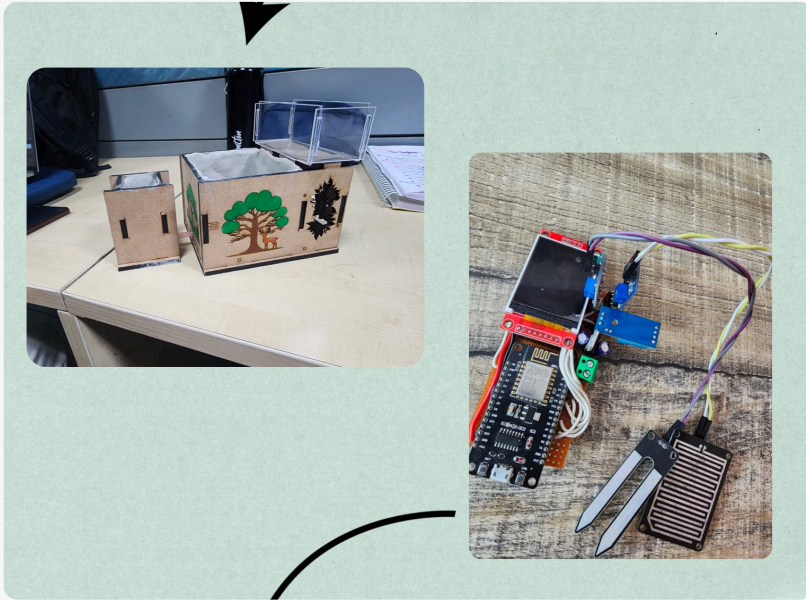
AutoCAD allows users to create detailed plans, layouts, and prototypes, and it provides features for precise measurement, scaling, and editing of designs. With its intuitive interface and powerful capabilities, AutoCAD is a popular choice among engineers, architects, designers, and hobbyists for creating accurate and professional-quality designs.



The Design of integration Box.

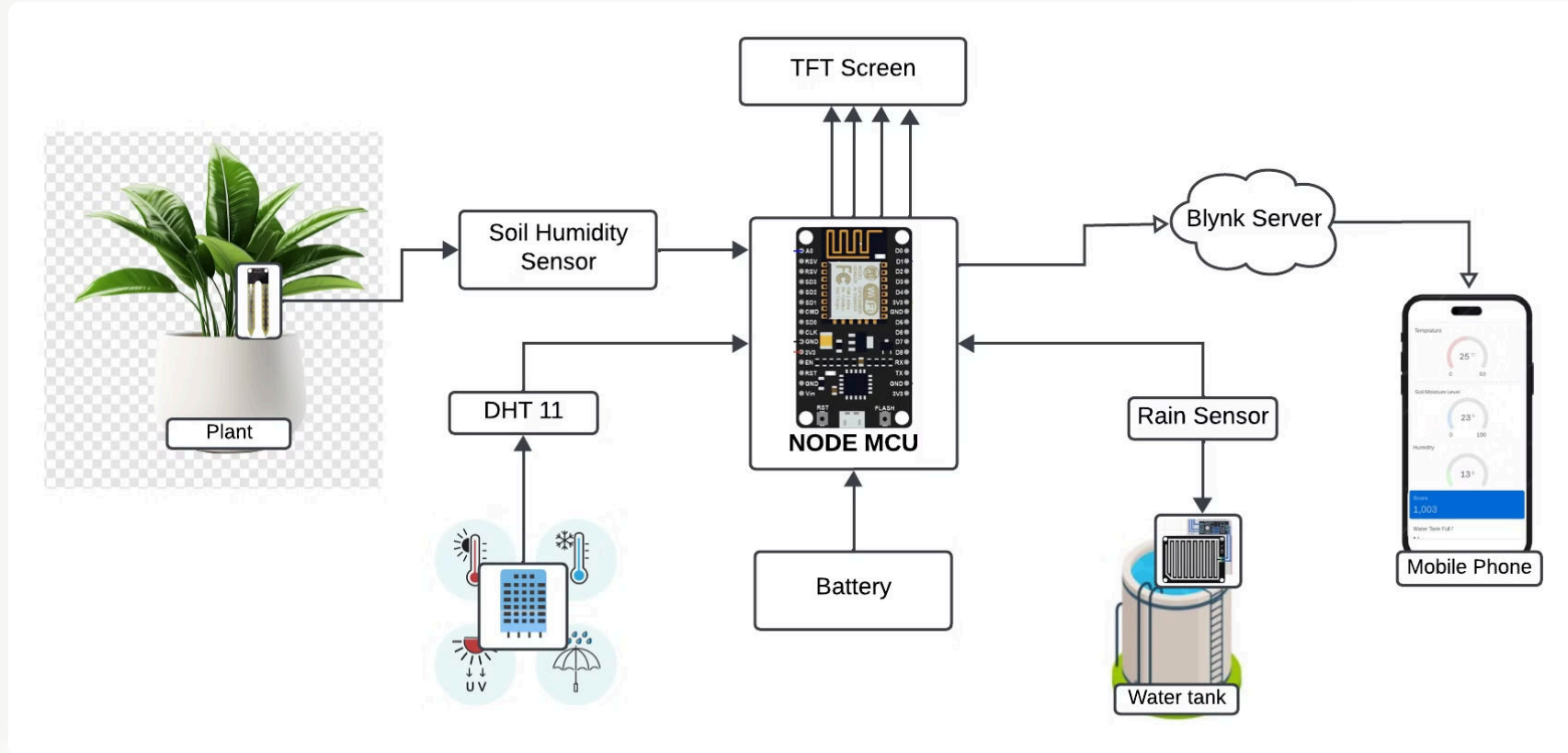


Final Box Look

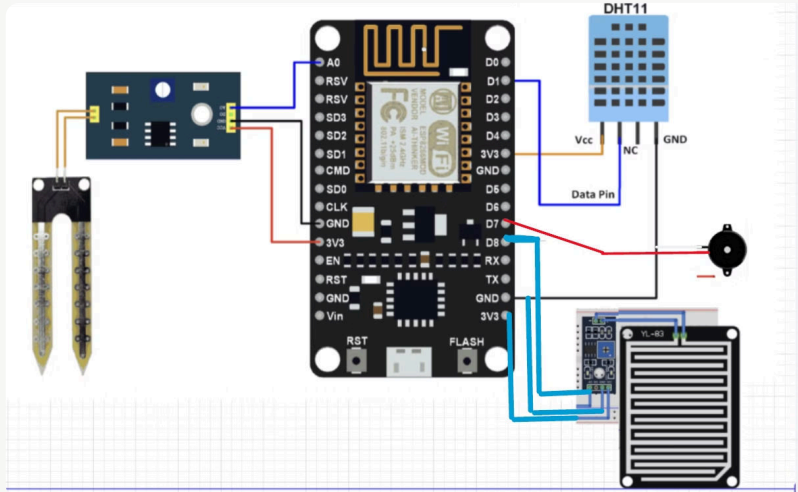


Electronics
Components and
Applying Clay on
box for water
proofing.

System Architecture

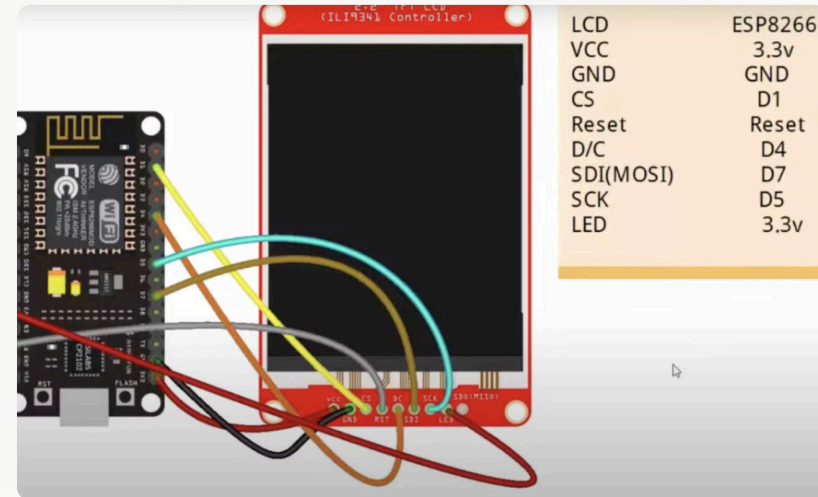


Pinouts To Make Connections



Connect DHT11, rain sensor in digital pins. Provide voltage through battery, add voltage step down bug converter wherever needed to step down the voltage according to device, mcu, supply.

Connect Soil Humidity Sensor to analog pin of the NodeMcu.



For TFT display, we are using SPI communication so we need 4 lines to connect :
MISO,MOSI,SC,SCLK.

Link for reference

(<https://arduino.stackexchange.com/questions/63133/how-to-connect-tft-lcd-display-with-nodemcu>)

Arduino IDE

code:<https://drive.google.com/drive/folders/1S62ak2Khb0cf2gZoCWotRsCLQegYBpGj?usp=sharing>

code: <https://drive.google.com/drive/folders/1S62ak2Khb0cf2gZoCWotRsCLQegYBpGj?usp=sharing>

```

41 #elif __cplusplus
42 int main() { RANK_AUTH_TOKEN; //Enter your Blynk web token
43 char ssid[] = "RabbitHole"; //Enter your WiFi name
44 char pass[] = "RabbitHole123"; //Enter your WiFi password
45
46 void setup() {
47   Serial.begin(115200);
48   pinMode(BUZZER, OUTPUT);
49   pinMode(LED, OUTPUT);
50   Blynk.config(auth, ssid, pass, "BlynkCloud", 8081);
51   delay(1000);
52   VFS.begin(MOUNT_POINT); // Init SDF7730 chip, black fat
53 }
54
55 void loop() {
56   Blynk.run(); // Run the Blynk function
57   digitalWrite(LED, LOW);
58
59   // Read Sensor Data
60   float ambientTempC();
61   float tempHumidityPercentage();
62   float humidityMoistureTOLMISTURE, SECS_R_H, SECS;
63   float humidRead(HUMIDITY);
64   if (temp == 0) {
65     Blynk.logEvent("temp_alert");
66     return -1;
67   }
68   digitalWrite(LED, HIGH);
69   digitalWrite(BUZZER, HIGH);
70   VFS.setCursor(0, 0);
71   VFS.println(SDF7730_MOUNT);
72   VFS.writeBytes(SDF7730_MOUNT);
73   VFS.setTimeOut(0);
74   VFS.printLine(0);
75   VFS.setTimeOut(20);
76   VFS.printLine(0);
77   VFS.printLine("Time");
78   VFS.printLine(0);
79   VFS.printLine("Full")
80
81   delay(1000);
82   digitalWrite(LED, LOW);
83   digitalWrite(BUZZER, LOW);
84 }
85
86 #else
87 #endif

```

```

1 # Final build project code
2 #
3 # Define the project name and version
4 #
5 # Define the project name and version
6 #
7 # Define the project name and version
8 #
9 # Define the project name and version
10 #
11 # Define the project name and version
12 #
13 # Define the project name and version
14 #
15 # Define the project name and version
16 #
17 # Define the project name and version
18 #
19 # Define the project name and version
20 #
21 # Define the project name and version
22 #
23 # Define the project name and version
24 #
25 # Define the project name and version
26 #
27 # Define the project name and version
28 #
29 # Define the project name and version
30 #
31 # Define the project name and version
32 #
33 # Define the project name and version
34 #
35 # Define the project name and version
36 #
37 # Define the project name and version
38 #
39 # Define the project name and version
40 #
41 # Define the project name and version
42 #
43 # Define the project name and version
44 #
45 # Define the project name and version
46 #
47 # Define the project name and version
48 #
49 # Define the project name and version
50 #
51 # Define the project name and version
52 #
53 # Define the project name and version
54 #
55 # Define the project name and version
56 #
57 # Define the project name and version
58 #
59 # Define the project name and version
60 #
61 # Define the project name and version
62 #
63 # Define the project name and version
64 #
65 # Define the project name and version
66 #
67 # Define the project name and version
68 #
69 # Define the project name and version
70 #
71 # Define the project name and version
72 #
73 # Define the project name and version
74 #
75 # Define the project name and version
76 #
77 # Define the project name and version
78 #
79 # Define the project name and version
80 #
81 # Define the project name and version
82 #
83 # Define the project name and version
84 #
85 # Define the project name and version
86 #
87 # Define the project name and version
88 #
89 # Define the project name and version
90 #
91 # Define the project name and version
92 #
93 # Define the project name and version
94 #
95 # Define the project name and version
96 #
97 # Define the project name and version
98 #
99 # Define the project name and version
100 #

```

[illegible]

```

Final_Aest_project_code.cpp
114
115 // Serial communication
116 Serial.print("Temperature : ");
117 Serial.println();
118 Serial.print(" Humidity : ");
119 Serial.println();
120 Serial.print(" Tank Full : ");
121 Serial.println();
122 Serial.print(" Soil Moisture Level : ");
123 Serial.println();
124
125
126 if(t > 250)
127 {
128     digitalWrite(temp_alert);
129 }
130 delay(500);
131 digitalWrite(0);
132
133
134 void TFTPrintText1() {
135     TFT.setCursor(10,10);
136     TFT.fillScreen(ST7735_YELLOW);
137     TFT.setCursor(10, 30);
138     TFT.setTextColor(ST7735_RED);
139     TFT.setTextSize(2);
140     TFT.setCursor(10,40);
141     TFT.print("Plant");
142     TFT.setCursor("Temperature :");
143     TFT.print("Temperature :");
144     TFT.print(" ");
145     TFT.setCursor(10,60);
146     delay(500);
147     TFT.fillScreen(ST7735_BLACK);
148     TFT.setCursor(10, 80);
149     TFT.setTextColor(ST7735_BLUE);
150     TFT.setTextSize(2);
151     TFT.setCursor(10,90);
152     TFT.print("Plant");
153     TFT.setCursor("Humidity :");
154     TFT.print("Humidity :");
155     TFT.setCursor(10,110);
156     delay(500);
157     TFT.setCursor(10, 130);
158     TFT.fillScreen(ST7735_WHITE);
159     TFT.setTextColor(ST7735_BLACK);
160     TFT.setTextSize(2);
161 }

```

```

Final_project_code.c
172 tfr.setCursor(0,1);
173 tfr.print("null?");
174 tfr.setCursor(0,1);
175 tfr.print("Maximum?");
176 tfr.setCursor(0,1);
177 tfr.print("Level : ");
178 tfr.setCursor(0,1);
179 tfr.print(a1);
180 delay(1000);
181 tfr.setCursor(0,3);
182 tfr.fillScreen(ST7735_RED);
183 tfr.setCursor(0,3);
184 tfr.print("Level : ");
185 tfr.setCursor(0,3);
186 tfr.print("Water?");
187 tfr.setCursor(0,3);
188 tfr.print("a2");
189 tfr.setCursor(0,3);
190 tfr.print(a2);
191 delay(1000);
192 tfr.setCursor(0,3);
193 tfr.fillScreen(ST7735_GREEN);
194 tfr.setCursor(0,3);
195 tfr.print("Level : ");
196 tfr.setCursor(0,3);
197 tfr.print("Temperature?");
198 tfr.setCursor(0,3);
199 tfr.print("a3");
200 tfr.setCursor(0,3);
201 tfr.print(a3);
202 delay(1000);
203 tfr.setCursor(0,3);
204 tfr.fillScreen(ST7735_BLACK);
205 tfr.setCursor(0,3);
206 tfr.print("Level : ");
207 tfr.setCursor(0,3);
208 tfr.print("Humidity?");
209 tfr.setCursor(0,3);
210 tfr.print("a4");
211 tfr.setCursor(0,3);
212 tfr.print(a4);
213 tfr.setCursor(0,3);
214 tfr.print("a5");
215 tfr.setCursor(0,3);
216 tfr.print("a6");
217 tfr.setCursor(0,3);
218 tfr.print("a7");
219 tfr.setCursor(0,3);
220 tfr.print("a8");
221 tfr.setCursor(0,3);
222 tfr.print("a9");
223 tfr.setCursor(0,3);
224 tfr.print("a10");
225 tfr.setCursor(0,3);
226 tfr.print("a11");
227 tfr.setCursor(0,3);
228 tfr.print("a12");
229 tfr.setCursor(0,3);
230 tfr.print("a13");
231 tfr.setCursor(0,3);
232 tfr.print("a14");
233 tfr.setCursor(0,3);
234 tfr.print("a15");
235 tfr.setCursor(0,3);
236 tfr.print("a16");
237 tfr.setCursor(0,3);
238 tfr.print("a17");
239 tfr.setCursor(0,3);
240 tfr.print("a18");
241 tfr.setCursor(0,3);
242 tfr.print("a19");
243 tfr.setCursor(0,3);
244 tfr.print("a20");
245 tfr.setCursor(0,3);
246 tfr.print("a21");
247 tfr.setCursor(0,3);
248 tfr.print("a22");
249 tfr.setCursor(0,3);
250 tfr.print("a23");
251 tfr.setCursor(0,3);
252 tfr.print("a24");
253 tfr.setCursor(0,3);
254 tfr.print("a25");
255 tfr.setCursor(0,3);
256 tfr.print("a26");
257 tfr.setCursor(0,3);
258 tfr.print("a27");
259 tfr.setCursor(0,3);
260 tfr.print("a28");
261 tfr.setCursor(0,3);
262 tfr.print("a29");
263 tfr.setCursor(0,3);
264 tfr.print("a30");
265 tfr.setCursor(0,3);
266 tfr.print("a31");
267 tfr.setCursor(0,3);
268 tfr.print("a32");
269 tfr.setCursor(0,3);
270 tfr.print("a33");
271 tfr.setCursor(0,3);
272 tfr.print("a34");
273 tfr.setCursor(0,3);
274 tfr.print("a35");
275 tfr.setCursor(0,3);
276 tfr.print("a36");
277 tfr.setCursor(0,3);
278 tfr.print("a37");
279 tfr.setCursor(0,3);
280 tfr.print("a38");
281 tfr.setCursor(0,3);
282 tfr.print("a39");
283 tfr.setCursor(0,3);
284 tfr.print("a40");
285 tfr.setCursor(0,3);
286 tfr.print("a41");
287 tfr.setCursor(0,3);
288 tfr.print("a42");
289 tfr.setCursor(0,3);
290 tfr.print("a43");
291 tfr.setCursor(0,3);
292 tfr.print("a44");
293 tfr.setCursor(0,3);
294 tfr.print("a45");
295 tfr.setCursor(0,3);
296 tfr.print("a46");
297 tfr.setCursor(0,3);
298 tfr.print("a47");
299 tfr.setCursor(0,3);
300 tfr.print("a48");
301 tfr.setCursor(0,3);
302 tfr.print("a49");
303 tfr.setCursor(0,3);
304 tfr.print("a50");
305 tfr.setCursor(0,3);
306 tfr.print("a51");
307 tfr.setCursor(0,3);
308 tfr.print("a52");
309 tfr.setCursor(0,3);
310 tfr.print("a53");
311 tfr.setCursor(0,3);
312 tfr.print("a54");
313 tfr.setCursor(0,3);
314 tfr.print("a55");
315 tfr.setCursor(0,3);
316 tfr.print("a56");
317 tfr.setCursor(0,3);
318 tfr.print("a57");
319 tfr.setCursor(0,3);
320 tfr.print("a58");
321 tfr.setCursor(0,3);
322 tfr.print("a59");
323 tfr.setCursor(0,3);
324 tfr.print("a60");
325 tfr.setCursor(0,3);
326 tfr.print("a61");
327 tfr.setCursor(0,3);
328 tfr.print("a62");
329 tfr.setCursor(0,3);
330 tfr.print("a63");
331 tfr.setCursor(0,3);
332 tfr.print("a64");
333 tfr.setCursor(0,3);
334 tfr.print("a65");
335 tfr.setCursor(0,3);
336 tfr.print("a66");
337 tfr.setCursor(0,3);
338 tfr.print("a67");
339 tfr.setCursor(0,3);
340 tfr.print("a68");
341 tfr.setCursor(0,3);
342 tfr.print("a69");
343 tfr.setCursor(0,3);
344 tfr.print("a70");
345 tfr.setCursor(0,3);
346 tfr.print("a71");
347 tfr.setCursor(0,3);
348 tfr.print("a72");
349 tfr.setCursor(0,3);
350 tfr.print("a73");
351 tfr.setCursor(0,3);
352 tfr.print("a74");
353 tfr.setCursor(0,3);
354 tfr.print("a75");
355 tfr.setCursor(0,3);
356 tfr.print("a76");
357 tfr.setCursor(0,3);
358 tfr.print("a77");
359 tfr.setCursor(0,3);
360 tfr.print("a78");
361 tfr.setCursor(0,3);
362 tfr.print("a79");
363 tfr.setCursor(0,3);
364 tfr.print("a80");
365 tfr.setCursor(0,3);
366 tfr.print("a81");
367 tfr.setCursor(0,3);
368 tfr.print("a82");
369 tfr.setCursor(0,3);
370 tfr.print("a83");
371 tfr.setCursor(0,3);
372 tfr.print("a84");
373 tfr.setCursor(0,3);
374 tfr.print("a85");
375 tfr.setCursor(0,3);
376 tfr.print("a86");
377 tfr.setCursor(0,3);
378 tfr.print("a87");
379 tfr.setCursor(0,3);
380 tfr.print("a88");
381 tfr.setCursor(0,3);
382 tfr.print("a89");
383 tfr.setCursor(0,3);
384 tfr.print("a90");
385 tfr.setCursor(0,3);
386 tfr.print("a91");
387 tfr.setCursor(0,3);
388 tfr.print("a92");
389 tfr.setCursor(0,3);
390 tfr.print("a93");
391 tfr.setCursor(0,3);
392 tfr.print("a94");
393 tfr.setCursor(0,3);
394 tfr.print("a95");
395 tfr.setCursor(0,3);
396 tfr.print("a96");
397 tfr.setCursor(0,3);
398 tfr.print("a97");
399 tfr.setCursor(0,3);
400 tfr.print("a98");
401 tfr.setCursor(0,3);
402 tfr.print("a99");
403 tfr.setCursor(0,3);
404 tfr.print("a100");
405 tfr.setCursor(0,3);
406 tfr.print("a101");
407 tfr.setCursor(0,3);
408 tfr.print("a102");
409 tfr.setCursor(0,3);
410 tfr.print("a103");
411 tfr.setCursor(0,3);
412 tfr.print("a104");
413 tfr.setCursor(0,3);
414 tfr.print("a105");
415 tfr.setCursor(0,3);
416 tfr.print("a106");
417 tfr.setCursor(0,3);
418 tfr.print("a107");
419 tfr.setCursor(0,3);
420 tfr.print("a108");
421 tfr.setCursor(0,3);
422 tfr.print("a109");
423 tfr.setCursor(0,3);
424 tfr.print("a110");
425 tfr.setCursor(0,3);
426 tfr.print("a111");
427 tfr.setCursor(0,3);
428 tfr.print("a112");
429 tfr.setCursor(0,3);
430 tfr.print("a113");
431 tfr.setCursor(0,3);
432 tfr.print("a114");
433 tfr.setCursor(0,3);
434 tfr.print("a115");
435 tfr.setCursor(0,3);
436 tfr.print("a116");
437 tfr.setCursor(0,3);
438 tfr.print("a117");
439 tfr.setCursor(0,3);
440 tfr.print("a118");
441 tfr.setCursor(0,3);
442 tfr.print("a119");
443 tfr.setCursor(0,3);
444 tfr.print("a120");
445 tfr.setCursor(0,3);
446 tfr.print("a121");
447 tfr.setCursor(0,3);
448 tfr.print("a122");
449 tfr.setCursor(0,3);
450 tfr.print("a123");
451 tfr.setCursor(0,3);
452 tfr.print("a124");
453 tfr.setCursor(0,3);
454 tfr.print("a125");
455 tfr.setCursor(0,3);
456 tfr.print("a126");
457 tfr.setCursor(0,3);
458 tfr.print("a127");
459 tfr.setCursor(0,3);
460 tfr.print("a128");
461 tfr.setCursor(0,3);
462 tfr.print("a129");
463 tfr.setCursor(0,3);
464 tfr.print("a130");
465 tfr.setCursor(0,3);
466 tfr.print("a131");
467 tfr.setCursor(0,3);
468 tfr.print("a132");
469 tfr.setCursor(0,3);
470 tfr.print("a133");
471 tfr.setCursor(0,3);
472 tfr.print("a134");
473 tfr.setCursor(0,3);
474 tfr.print("a135");
475 tfr.setCursor(0,3);
476 tfr.print("a136");
477 tfr.setCursor(0,3);
478 tfr.print("a137");
479 tfr.setCursor(0,3);
480 tfr.print("a138");
481 tfr.setCursor(0,3);
482 tfr.print("a139");
483 tfr.setCursor(0,3);
484 tfr.print("a140");
485 tfr.setCursor(0,3);
486 tfr.print("a141");
487 tfr.setCursor(0,3);
488 tfr.print("a142");
489 tfr.setCursor(0,3);
490 tfr.print("a143");
491 tfr.setCursor(0,3);
492 tfr.print("a144");
493 tfr.setCursor(0,3);
494 tfr.print("a145");
495 tfr.setCursor(0,3);
496 tfr.print("a146");
497 tfr.setCursor(0,3);
498 tfr.print("a147");
499 tfr.setCursor(0,3);
500 tfr.print("a148");
501 tfr.setCursor(0,3);
502 tfr.print("a149");
503 tfr.setCursor(0,3);
504 tfr.print("a150");
505 tfr.setCursor(0,3);
506 tfr.print("a151");
507 tfr.setCursor(0,3);
508 tfr.print("a152");
509 tfr.setCursor(0,
```

The architecture of the Smart IoT Botany STEM Kit :

1. Hardware Components:

- **Microcontroller Unit (MCU):** The MCU serves as the brain of the system, responsible for gathering data from sensors, processing information, and controlling output devices.
- **Sensors:** Soil moisture sensor, water level sensor, and temperature & humidity sensor collect data related to plant health and environmental conditions.
- **Display Screen:** An LCD display screen provides visual feedback to users, such as current soil moisture level, temperature, and humidity readings.
- **Enclosure:** The MDF board and acrylic panels form an enclosure to house the electronic components and protect them from external elements.
- **Power Supply:** Batteries or a power adapter provide the necessary power to operate the system.

2. Software Components:

- **Arduino IDE:** Used for programming the MCU to read sensor data, process it, and control output devices based on predefined logic.
- **Blynk App:** An IoT platform used for remote monitoring and control of the Smart IoT Botany STEM Kit. It allows users to view real-time sensor data and adjust settings from a smartphone or tablet.

3. Data Flow:

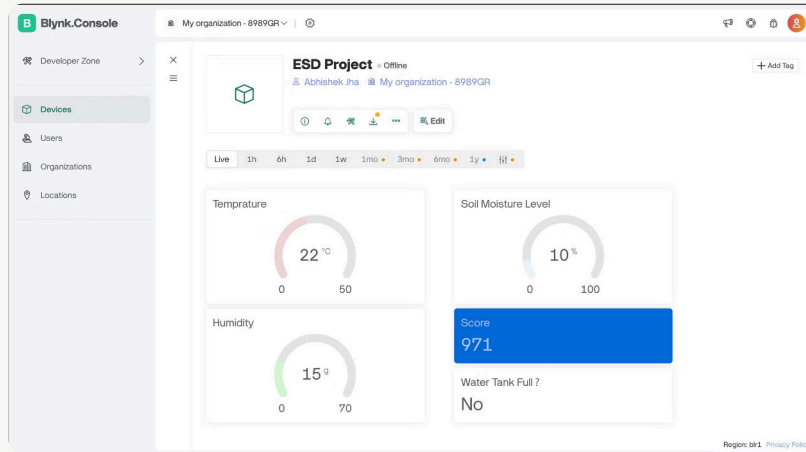
- The sensors continuously monitor plant health parameters such as soil moisture level, water level in the tank, temperature, and humidity.
- Sensor data is collected by the MCU, which processes the information and determines if any actions need to be taken (e.g., watering the plant).
- The MCU communicates with the Blynk app via Wi-Fi, sending sensor data and receiving commands from the user.
- Users can view real-time sensor data and adjust settings (e.g., watering schedule) using the Blynk app.
- Based on user input and sensor readings, the MCU controls output devices such as the display screen to provide feedback and alerts.

4. Integration:

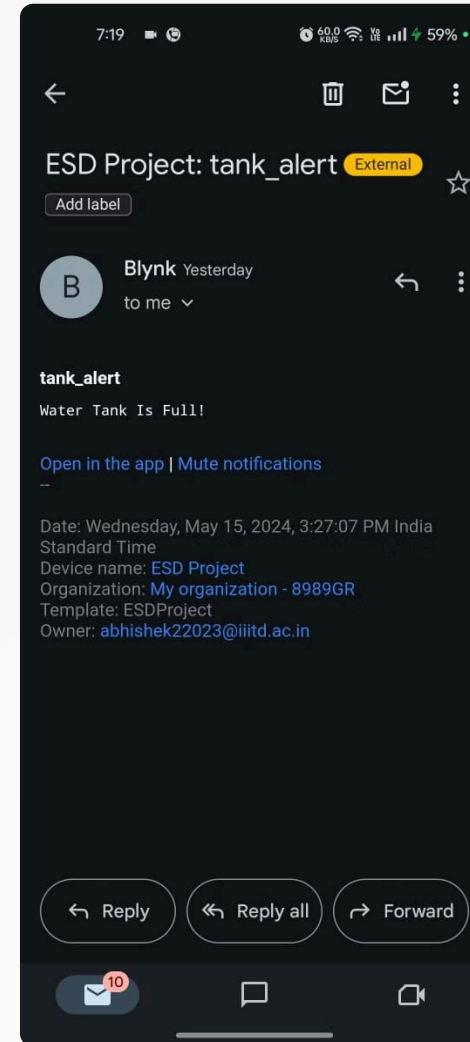
- The hardware and software components are integrated to create a cohesive system that enables users to monitor and control plant growth parameters remotely.
- The MCU serves as the central hub, orchestrating the flow of data between sensors, the Blynk app, and output devices.
- The enclosure provides physical protection for the electronic components while allowing for easy access and visibility of the display screen.

Overall, the architecture of the Smart IoT Botany STEM Kit is designed to be user-friendly, flexible, and reliable, enabling users to engage in hands-on botany and STEM learning experiences while leveraging IoT technology for plant care.

Blynk Software

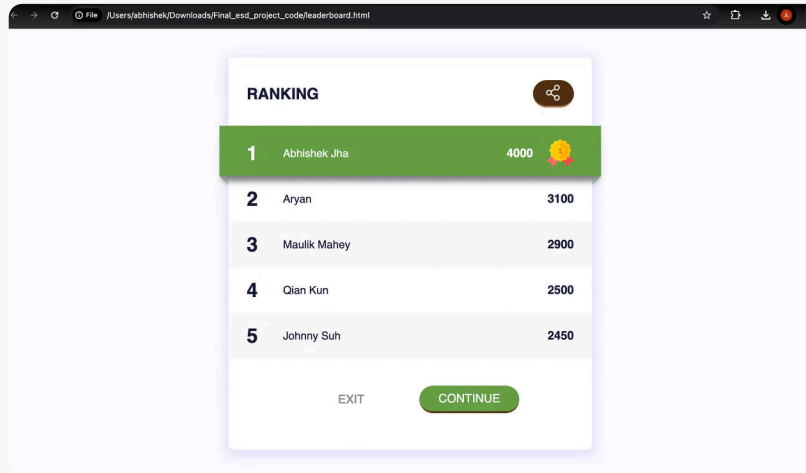


This is the real time update of data and score of the plant. Includes temperature , humidity and moisture level and Tank details. Score is a scaled level of all 3 above.



You will get real time notifications if everything is good or not.

You can also use blynk app to get al data on your mobile.



Link to leaderboard:

Gamification

The leaderboard and streak elements of the Smart IoT Botany STEM Kit gamify the plant care experience, adding fun and motivation for users. The leaderboard lets users compete to achieve the best plant growth results, while the streak feature rewards consistent plant care routines. These elements encourage engagement, competition, and dedication to nurturing plants, making the learning experience more interactive and rewarding.

IMPLEMENTATION DETAILS:

1. Assembly of Hardware Components:

- Begin by assembling the hardware components onto the MDF board. This includes mounting the microcontroller unit (MCU), sensors (soil moisture, water level, temperature & humidity), and the LCD display screen.
- Ensure that the components are securely attached and properly connected using jumper wires or soldering, following the circuit diagrams and schematics.

2. Waterproofing the Enclosure:

- Apply MSEAL clay or similar waterproofing material to the enclosure to protect the electronic components from moisture and spills. This step is crucial to ensure the longevity and reliability of the Smart IoT Botany STEM Kit.

3. Programming the Microcontroller Unit (MCU):

- Use the Arduino IDE or compatible software to program the MCU. Write code to read data from the sensors, process it based on predefined thresholds, and control output devices accordingly.
- Implement logic to monitor soil moisture levels, water levels in the tank, and environmental conditions (temperature and humidity). Trigger alerts or actions (such as watering the plant) when necessary.

4. Configuring the Blynk App:

- Set up the Blynk app to communicate with the MCU via Wi-Fi. Create a user interface with widgets to display real-time sensor data and control settings.
- Establish communication protocols and authentication mechanisms to ensure secure and reliable data exchange between the app and the MCU.

5. Testing and Calibration:

- Test the Smart IoT Botany STEM Kit thoroughly to ensure that all components are functioning correctly. Verify sensor accuracy and responsiveness under different environmental conditions.
- Calibrate sensors if necessary to ensure accurate readings and optimal performance.

6. User Manual and Documentation:

- Prepare a user manual that provides detailed instructions on how to assemble, set up, and operate the Smart IoT Botany STEM Kit. Include troubleshooting tips and FAQs to assist users in resolving common issues.
- Document the implementation process, including circuit diagrams, code snippets, and configuration settings, for future reference and troubleshooting.

7. Packaging and Branding:

- Design packaging materials (box, labels, etc.) for the Smart IoT Botany STEM Kit to create a cohesive and professional presentation.
- Incorporate branding elements (logo, colors, etc.) to establish a recognizable identity for the product and enhance its appeal to users.

User Guide of the product.

Assembly Guide for Smart IoT Botany STEM Kit

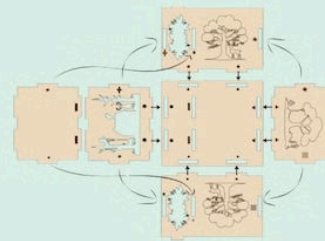
1 Unboxing

Remove all components from the packaging and ensure that you have everything needed:

- **MDF Board** (prototyping base)
- **MSEAL Clay** (waterproofing)
- **Acrylic Panels** (transparency)
- **Sensors** (**water level sensor**, **soil moisture sensor**, **temperature & humidity sensor**)
- **Electronic components** (**Arduino board**, **buzzer**, **LED indicators**, **LCD Screen**)
- **Water tank** (for excess water collection)

2 Joining the MDF Planks

- Make Sure to align the Shapes printed on the MDF planks with their matching shapes and join them by putting the cutouts into the slits.
- For better Demonstration, please refer to the Illustration beside.



3 Waterproofing

Apply MSEAL clay to the enclosure to waterproof the kit. Ensure that all seams and edges are properly sealed to protect the electronic components from moisture and spills. Allow the clay to dry completely before proceeding to the next step.

4 Plant Placement

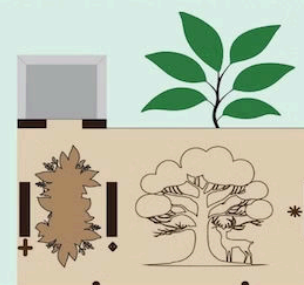
Carefully place your favorite plant into the designated planting area on the MDF board.

5 Electronic Component Setup

Carefully place the already prepared Circuit using Breadboard, Sensors and Chipsets into the Empty side compartment.

6 Final Assembly and App Setup

Secure all components in place and double-check that everything is properly aligned and connected. Download the companion app on your smartphone and follow the setup instructions to connect to the Smart IoT Botany STEM Kit via Wi-Fi. Once connected, you can access real-time sensor data and control features directly from the app.



Demonstration Video:



My movie 1.mp4



Testing and Evaluation:

During the development of the Smart IoT Botany STEM Kit, rigorous testing procedures were employed to ensure the reliability, functionality, and safety of the system. Various tests were conducted to evaluate the performance of hardware components, verify sensor accuracy, and validate the effectiveness of the IoT platform. Here's an overview of the testing and evaluation process:

1. Voltage and Current Testing:

- Voltage and current measurements were taken at different points in the circuit to ensure that components were operating within their specified ranges and that power requirements were met.
- Testing included verifying the voltage supplied to the microcontroller unit (MCU) and sensors, as well as measuring the current drawn by the system during different operation modes.

2. Buck Converter Evaluation:

- The performance of the buck converter used to regulate voltage levels for the MCU and other components was evaluated.
- Efficiency tests were conducted to assess the converter's ability to step down voltage while minimizing power losses.
- Output voltage stability and ripple were measured to ensure a consistent and reliable power supply to the system.

3. Sensor Calibration and Accuracy Testing:

- Soil moisture, water level, temperature, and humidity sensors were calibrated against known standards to verify their accuracy and reliability.
- Sensor readings were compared to reference measurements to assess their precision and consistency across different environmental conditions.
- Calibration curves were generated to translate sensor readings into meaningful data points for plant monitoring and control.

4. IoT Platform Testing:

- The Blynk app interface was tested for compatibility with various smartphones and tablets, ensuring a seamless user experience across different devices.
- Communication protocols between the MCU and the Blynk server were evaluated to verify data transmission reliability and security.
- User interactions with the app, such as adjusting settings and viewing real-time sensor data, were tested to identify any usability issues or bugs.

5. End-to-End System Testing:

- Comprehensive end-to-end testing was conducted to validate the functionality and integration of hardware and software components.
- This included testing scenarios such as sensor data acquisition, alert triggering based on predefined thresholds, remote control via the Blynk app, and display of real-time data on the LCD screen.
- System performance was assessed under normal operating conditions as well as stress tests to identify any potential failure points or performance bottlenecks.

By conducting thorough testing and evaluation, the Smart IoT Botany STEM Kit was able to deliver a reliable, user-friendly, and effective solution for engaging teenage kids in botany and STEM learning while leveraging IoT technology for plant care.

Future Enhancements:

1. Integration of Telegram Bot for Enhanced User Interface:

- In future iterations of the Smart IoT Botany STEM Kit, we plan to integrate a Telegram bot as an alternative user interface for even greater accessibility and ease of use.
- The Telegram bot will allow users to monitor their plants, receive notifications, and adjust settings directly from their smartphones using the Telegram messaging platform.
- This additional interface will provide users with more flexibility and convenience in interacting with the Smart IoT Botany STEM Kit, catering to diverse preferences and technological capabilities.

2. Automated Threshold Setting Algorithm for Optimal Plant Care:

- To further streamline the plant care process, we aim to develop an automated algorithm that dynamically sets the required thresholds for water level, humidity, and temperature based on specific plant requirements.
- This algorithm will utilize machine learning techniques and plant database information to analyze historical sensor data and determine optimal environmental conditions for different types of plants.
- By removing the need for manual input and calibration, this automated approach will simplify setup and maintenance tasks for users, ensuring that plants receive optimal care without requiring constant monitoring or intervention.
- Additionally, the algorithm will continuously adapt and refine its recommendations over time, taking into account changing environmental factors and plant growth stages to optimize long-term plant health and productivity.

These future enhancements will further elevate the Smart IoT Botany STEM Kit, making it more user-friendly, intelligent, and effective in fostering botanical learning and environmental stewardship among teenage users.

Challenges Faced:

1. **Voltage Conversion:** Voltage compatibility between components can present challenges, particularly when different components require varying voltage levels for optimal operation. Ensuring proper voltage conversion while maintaining current requirements is crucial to prevent damage to components. In our project, the utilization of buck converters facilitated voltage regulation without affecting the current, addressing voltage conversion challenges effectively.
2. **Power Supply Design:** Designing a stable and efficient power supply system can be challenging, especially when powering multiple components with diverse voltage and current requirements. Challenges such as voltage regulation, noise suppression, and power efficiency optimization must be addressed to meet the project's power supply needs reliably.
3. **Component Soldering and Assembly:** Proper soldering and assembly practices are essential for robust connections and reliable circuit operation. Challenges may arise in ensuring high-quality solder joints, precise component placement, and mechanical stability, particularly in small-scale or densely packed circuits.
4. **Testing and Debugging:** Identifying and resolving issues in electronic circuits through testing and debugging can be time-consuming and challenging. Comprehensive testing procedures, including functional testing, signal analysis, and troubleshooting techniques, are necessary to validate circuit performance and diagnose problems effectively.
5. **Component Limitations and Trade-offs:** Working within the constraints of available resources, such as limited pins or memory, can pose challenges in component selection and circuit design. Making informed decisions and balancing trade-offs based on project requirements are essential to overcome such limitations effectively. For example, in our project, the use of a rain sensor instead of a water level sensor due to pin constraints on the NodeMCU required careful consideration of sensing capabilities and trade-offs in functionality.

Obstacles

Output pins of sensors give 5V output but nodeMCU was taking 3.3v input for safety precautions.



Buck Converter

Solutions

We used 5V→3V Voltage converters. They work on the principle of

1. $U_1/U_2 = N_1/N_2$, that is, for any two coils of the same transformer, the voltage is proportional to the number of turns.
2. $P_{in} = P_{out}$, that is, no matter how many secondary coils are working, the input power of the transformer is always equal to the sum of all output powers.

Bill Of Materials and Purchase Links:

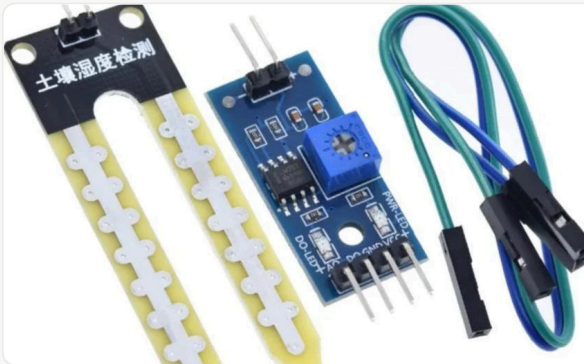
1

Microcontroller Unit (MCU):

- NodeMCU(ESP 8266) | RS.151/-

2

Soil Moisture Sensor| RS.74/-



Robocraze



Soil Moisture Sensor Module

Soil Moisture Sensor Module This is Soil Moisture Sensor Module or Soil Moisture Meter, Soil Humidity Sensor, Water...

3

Rain Sensor| RS.48/-



Robu.in | Indian Online Store | RC Hobby | Robotics

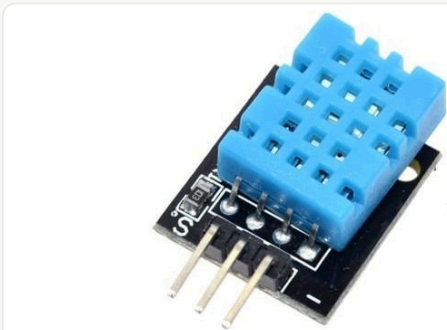


Buy Raindrops Detection Sensor Module Online at R...

Upgrade your weather station with our Raindrops Detection Sensor Module. Real-time rain monitoring made easy. Buy...

4

DHT 11| RS.49/-



Robu.in | Indian Online Store | RC Hobby | Robotics



Buy DHT11 Digital Relative Humidity and Temperatu...

Stay Informed with DHT11 Digital Relative Humidity and Temperature Sensor Module. Perfect for IoT & DIY projects....

5

TFT display| RS.499/-



Robu.in | Indian Online Store | RC Hobby | Robotics



Buy 2.4-inch SPI Interface 240x320 TFT Display M...

Buy 2.4-inch SPI Interface 240x320 TFT Display Module online at lowest price at Robu.in. Avail free shipping all over...

6

Voltage Step Down Converter| RS.35/-



Robu.in | Indian Online Store | RC Hobby | Robotics



Buy DC-DC LM2596 Buck Converter Step down Mo...

Get the LM2596 Buck Converter?is a handy power accessory board that allows you to fine tune the amount of...

Submitted To:

Course Instructor:

Mr. Alok Mitta(alok.mittal@st.com)

TA: Sandeep Kumar. (sandeep23186@iiitd.ac.in)

TA: Ashutosh (ashutosh23160@iiitd.ac.in)