**Algorithm:**

1. Soil Moisture Sensing Algorithm:

if soil\_moisture < moisture\_threshold:

activate\_watering\_system()

1. Watering Duration Algorithm:

watering\_duration = calculate\_watering\_duration()

activate\_watering\_system(duration=watering\_duration)

1. Weather-based Adjustment Algorithm:

if is\_dry\_season() and no\_rainfall\_forecasted():

adjust\_watering\_schedule()

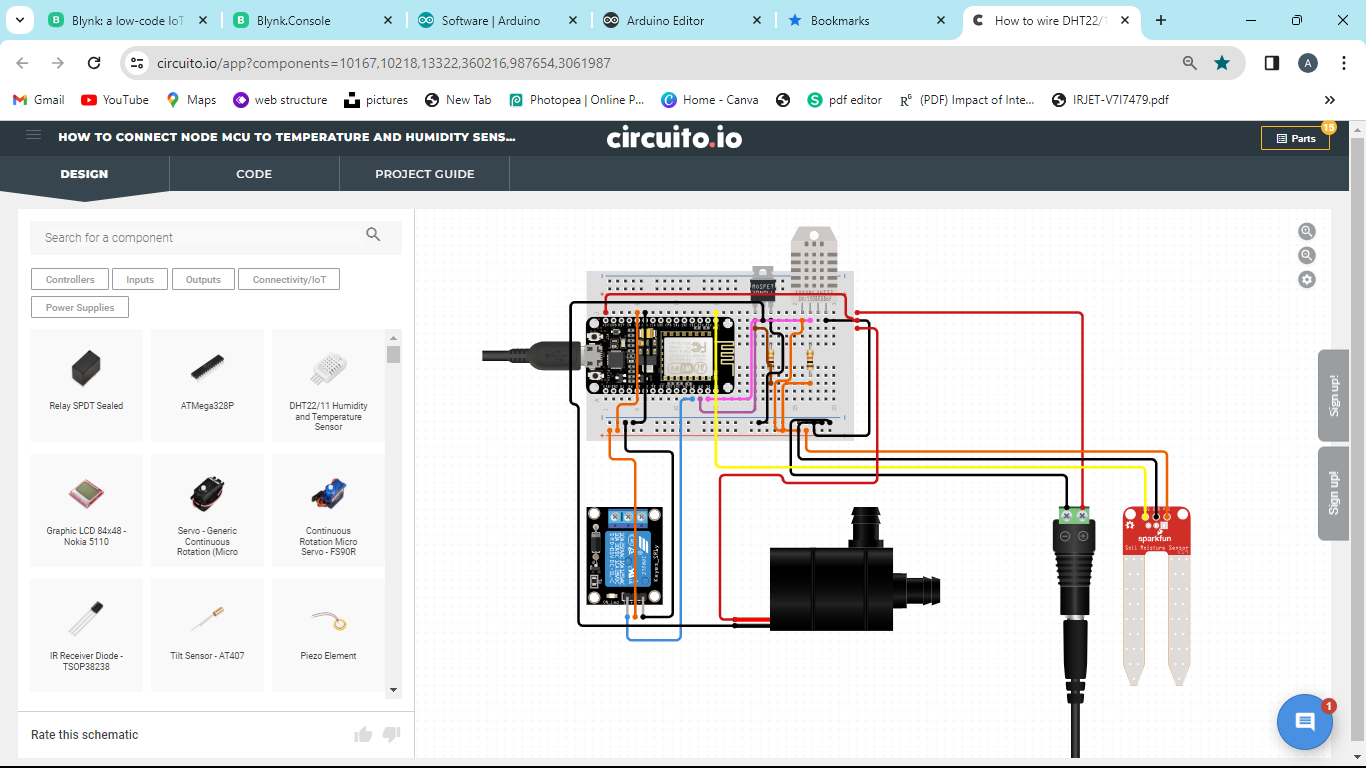
1. Blynk Integration Algorithm:

blynk.virtual\_write(V1, soil\_moisture)

if blynk\_button\_pressed():

manual\_watering\_mode()

**Online Simulation:**

****

**Coding:**

|  |
| --- |
| // Include Libraries  #include "Arduino.h"  #include "DHT.h"  #include "Relay.h"  #include "SoilMoisture.h"  #include "Pump.h"  // Pin Definitions  #define DHT\_PIN\_DATA 5  #define RELAYMODULE\_PIN\_SIGNAL 0  #define SOILMOISTURE\_3V3\_PIN\_SIG A0  #define WATERPUMP\_PIN\_COIL1 4  // Global variables and defines  // object initialization  DHT dht(DHT\_PIN\_DATA);  Relay relayModule(RELAYMODULE\_PIN\_SIGNAL);  SoilMoisture soilMoisture\_3v3(SOILMOISTURE\_3V3\_PIN\_SIG);  Pump waterpump(WATERPUMP\_PIN\_COIL1);  // define vars for testing menu  const int timeout = 10000; //define timeout of 10 sec  char menuOption = 0;  long time0;  // Setup the essentials for your circuit to work. It runs first every time your circuit is powered with electricity.  void setup()  {  // Setup Serial which is useful for debugging  // Use the Serial Monitor to view printed messages  Serial.begin(9600);  while (!Serial) ; // wait for serial port to connect. Needed for native USB  Serial.println("start");  dht.begin();  menuOption = menu();  }  // Main logic of your circuit. It defines the interaction between the components you selected. After setup, it runs over and over again, in an eternal loop.  void loop()  {  if(menuOption == '1') {  // DHT22/11 Humidity and Temperature Sensor - Test Code  // Reading humidity in %  float dhtHumidity = dht.readHumidity();  // Read temperature in Celsius, for Fahrenheit use .readTempF()  float dhtTempC = dht.readTempC();  Serial.print(F("Humidity: ")); Serial.print(dhtHumidity); Serial.print(F(" [%]\t"));  Serial.print(F("Temp: ")); Serial.print(dhtTempC); Serial.println(F(" [C]"));  }  else if(menuOption == '2') {  // Relay Module - Test Code  // The relay will turn on and off for 500ms (0.5 sec)  relayModule.on(); // 1. turns on  delay(500); // 2. waits 500 milliseconds (0.5 sec). Change the value in the brackets (500) for a longer or shorter delay in milliseconds.  relayModule.off(); // 3. turns off.  delay(500); // 4. waits 500 milliseconds (0.5 sec). Change the value in the brackets (500) for a longer or shorter delay in milliseconds.  }  else if(menuOption == '3') {  // Soil Moisture Sensor - Test Code  int soilMoisture\_3v3Val = soilMoisture\_3v3.read();  Serial.print(F("Val: ")); Serial.println(soilMoisture\_3v3Val);  }  else if(menuOption == '4') {  // Submersible Pool Water Pump - Test Code  // The water pump will turn on and off for 2000ms (4 sec)  waterpump.on(); // 1. turns on  delay(2000); // 2. waits 500 milliseconds (0.5 sec).  waterpump.off();// 3. turns off  delay(2000); // 4. waits 500 milliseconds (0.5 sec).  }  if (millis() - time0 > timeout)  {  menuOption = menu();  }    }  // Menu function for selecting the components to be tested  // Follow serial monitor for instrcutions  char menu()  {  Serial.println(F("\nWhich component would you like to test?"));  Serial.println(F("(1) DHT22/11 Humidity and Temperature Sensor"));  Serial.println(F("(2) Relay Module"));  Serial.println(F("(3) Soil Moisture Sensor"));  Serial.println(F("(4) Submersible Pool Water Pump"));  Serial.println(F("(menu) send anything else or press on board reset button\n"));  while (!Serial.available());  // Read data from serial monitor if received  while (Serial.available())  {  char c = Serial.read();  if (isAlphaNumeric(c))  {    if(c == '1')  Serial.println(F("Now Testing DHT22/11 Humidity and Temperature Sensor"));  else if(c == '2')  Serial.println(F("Now Testing Relay Module"));  else if(c == '3')  Serial.println(F("Now Testing Soil Moisture Sensor"));  else if(c == '4')  Serial.println(F("Now Testing Submersible Pool Water Pump"));  else  {  Serial.println(F("illegal input!"));  return 0;  }  time0 = millis();  return c;  }  }  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Circuito.io is an automatic generator of schematics and code for off  \* the shelf hardware combinations.  \* Copyright (C) 2016 Roboplan Technologies Ltd.  \* This program is free software: you can redistribute it and/or modify  \* it under the terms of the GNU General Public License as published by  \* the Free Software Foundation, either version 3 of the License, or  \* (at your option) any later version.  \* This program is distributed in the hope that it will be useful,  \* but WITHOUT ANY WARRANTY; without even the implied warranty of  \* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  \* GNU General Public License for more details.  \* You should have received a copy of the GNU General Public License  \* along with this program. If not, see <http://www.gnu.org/licenses/>.  \* In addition, and without limitation, to the disclaimers of warranties  \* stated above and in the GNU General Public License version 3 (or any  \* later version), Roboplan Technologies Ltd. ("Roboplan") offers this  \* program subject to the following warranty disclaimers and by using  \* this program you acknowledge and agree to the following:  \* THIS PROGRAM IS PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS, AND  \* WITHOUT WARRANTIES OF ANY KIND EITHER EXPRESS OR IMPLIED. ROBOPLAN  \* HEREBY DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT  \* NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, TITLE, FITNESS  \* FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, AND THOSE ARISING BY  \* STATUTE OR FROM A COURSE OF DEALING OR USAGE OF TRADE.  \* YOUR RELIANCE ON, OR USE OF THIS PROGRAM IS AT YOUR SOLE RISK.  \* ROBOPLAN DOES NOT GUARANTEE THAT THE PROGRAM WILL BE FREE OF, OR NOT  \* SUSCEPTIBLE TO, BUGS, SECURITY BREACHES, OR VIRUSES. ROBOPLAN DOES  \* NOT WARRANT THAT YOUR USE OF THE PROGRAM, INCLUDING PURSUANT TO  \* SCHEMATICS, INSTRUCTIONS OR RECOMMENDATIONS OF ROBOPLAN, WILL BE SAFE  \* FOR PERSONAL USE OR FOR PRODUCTION OR COMMERCIAL USE, WILL NOT  \* VIOLATE ANY THIRD PARTY RIGHTS, WILL PROVIDE THE INTENDED OR DESIRED  \* RESULTS, OR OPERATE AS YOU INTENDED OR AS MAY BE INDICATED BY ROBOPLAN.  \* YOU HEREBY WAIVE, AGREE NOT TO ASSERT AGAINST, AND RELEASE ROBOPLAN,  \* ITS LICENSORS AND AFFILIATES FROM, ANY CLAIMS IN CONNECTION WITH ANY OF  \* THE ABOVE.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/ |