

GREENHOUSE

DMP PROJECT



January 19, 2023

Ciobanu Dan-Dumitru

GROUP 30433, 3RD YEAR



**GREENHOUSE**

1. **SHORT DESCRIPTION**

The project is about creating sort of a greenhouse in the way that it would simulate the way water would be poured automatically when periodical measurements indicate the conditions for this action (at about every 10 seconds, the reading of the soil moisture is triggered by an internal interrupt). Also the user is able to pour the water at about any moment he wants with the help of an external interrupt by pressing a button (except for the moment when the servo motor is already working due to an internal interrupt). The user is also able to read the temperature in the air, the level in the water container and the humidity of the soil.

1. **COMPONENTS USED**
   1. **DHT11 Sensor for Measuring Temperature and Humidity in Air**
   2. **FUNDUINO Sensor for Measuring Water Level**
   3. **Sensor for Measuring the Soil Humidity**
   4. **Servo Motor**
   5. **Breadboard**
   6. **1 led**
   7. **1 10k Ohm Resistor**
   8. **1 220 Ohm Resistor**
   9. **Cables, Scotch, Cardboard**
2. **TECHNICAL ASPECTS**

As technical aspects worth mentioning, there could be mentioned: the use of internal interrupts and external interrupts, the use of the ‘Servo.h’ library for the servo motor.

The complicated thing that I figured out when implementing the project is that the use of interrupts could be inconvenient when using methods that deal with interrupts inside their implementation. I found out that there needs to be a delimitation in the logic in the code that would separate the interrupts one from another; detachments should be used before switching to another attachment of interrupt.

The sensors are being read with the help of ‘analogRead’ function from the analog ports: A0 and A1. The DHT11 sensor is a bit more complicated though. The pin used for it is the digital pin number 5. The method used for reading it was found on the internet as I did not manage to find any information that would hint to the way that the implementation should be made.

There is also a little led on the digital pin 13 that lights up whenever the water level drops below a certain value.

1. **EXTERNAL LINKS**
   1. [**https://electronicsprojectshub.com/setup-dht11-sensor-with-arduino/**](https://electronicsprojectshub.com/setup-dht11-sensor-with-arduino/)
   2. [**https://www.circuitbasics.com/arduino-basics-controlling-led/**](https://www.circuitbasics.com/arduino-basics-controlling-led/)
   3. [**https://www.engineersgarage.com/articles-arduino-dht11-humidity-temperature-sensor-interfacing/**](https://www.engineersgarage.com/articles-arduino-dht11-humidity-temperature-sensor-interfacing/)
   4. [**https://arduinogetstarted.com/tutorials/arduino-water-sensor**](https://arduinogetstarted.com/tutorials/arduino-water-sensor)
   5. [**https://lastminuteengineers.com/soil-moisture-sensor-arduino-tutorial/**](https://lastminuteengineers.com/soil-moisture-sensor-arduino-tutorial/)
2. **SOURCE CODE**
3. #include "Servo.h"
4. #include <TimerOne.h>
5. volatile Servo myServo;
6. volatile int needWater = 0;
7. volatile int soilAverage = 0;
8. volatile int pourSomeWater = 0;
9. volatile int waterLevel = 0;
10. #define WATER\_LEVEL\_SIGNAL\_PIN A0
11. #define WATER\_LEVEL\_POWER\_PIN 7
12. #define SOIL\_HUMIDITY\_SIGNAL\_PIN A1
13. #define SOIL\_HUMIDITY\_POWER\_PIN 6
14. #define SERVO\_CONTROL\_PIN 9
15. #define DHT\_PIN 5
16. #define LED\_PIN 2
17. #define INTERRUPT\_BUTTON 2
18. void setup() {
19. Serial.begin(9600);
20. pinMode(WATER\_LEVEL\_POWER\_PIN, OUTPUT);
21. pinMode(SOIL\_HUMIDITY\_POWER\_PIN, OUTPUT);
22. pinMode(DHT\_PIN, INPUT);
23. pinMode(INTERRUPT\_BUTTON, INPUT\_PULLUP);
24. digitalWrite(WATER\_LEVEL\_POWER\_PIN, LOW);
25. digitalWrite(SOIL\_HUMIDITY\_POWER\_PIN, LOW);
26. digitalWrite(SERVO\_CONTROL\_PIN, LOW);
27. digitalWrite(13, LOW);
28. Timer1.initialize(100000000);
29. Timer1.attachInterrupt(verify);
30. }
31. void loop() {
32. digitalWrite(SOIL\_HUMIDITY\_SIGNAL\_PIN, HIGH);
33. digitalWrite(WATER\_LEVEL\_POWER\_PIN, HIGH);
34. if (waterLevel < 500) {
35. digitalWrite(13, HIGH);
36. } else {
37. digitalWrite(13, LOW);
38. }
39. for (int i = 0; i < 5; i++) {
40. wait\_for\_dht11();
41. start\_signal(5);
42. read\_dht11(5);
43. }
44. if (needWater == 1 && waterLevel > 500) {
45. myServo.attach(SERVO\_CONTROL\_PIN);
46. Serial.println("Dry");
47. int currentIndex = 0;
48. int soilHumidity = 200;
49. for (int i = 0; i < 180 && soilHumidity > 190; i++) {
50. currentIndex++;
51. soilHumidity = analogRead(SOIL\_HUMIDITY\_SIGNAL\_PIN);
52. Serial.println(soilHumidity);
53. myServo.write(i);
54. delay(15);
55. }
56. for (int i = currentIndex; i >= 0; i--) {
57. myServo.write(i);
58. delay(15);
59. }
60. myServo.detach();
61. needWater = 0;
62. }
63. delay(1000);
64. attachInterrupt(digitalPinToInterrupt(INTERRUPT\_BUTTON), pourWater, RISING);
65. for (int i = 0; i < 100; i++) {
66. Serial.print("Water level: ");
67. Serial.println(analogRead(WATER\_LEVEL\_SIGNAL\_PIN));
68. delay(10);
69. }
70. detachInterrupt(digitalPinToInterrupt(INTERRUPT\_BUTTON));
71. if (pourSomeWater == 1) {
72. myServo.attach(SERVO\_CONTROL\_PIN);
73. for (int i = 0; i < 180; i++) {
74. myServo.write(i);
75. delay(15);
76. }
77. for (int i = 179; i >= 0; i--) {
78. myServo.write(i);
79. delay(15);
80. }
81. myServo.detach();
82. delay(50);
83. pourSomeWater = 0;
84. delay(50);
85. }
86. delay(1000);
87. Timer1.attachInterrupt(verify);
88. delay(10);
89. waterLevel = analogRead(WATER\_LEVEL\_SIGNAL\_PIN);
90. digitalWrite(WATER\_LEVEL\_POWER\_PIN, LOW);
91. digitalWrite(SOIL\_HUMIDITY\_POWER\_PIN, LOW);
92. delay(1000);
93. }
94. void pourWater() {
95. delay(50);
96. pourSomeWater = 1;
97. delay(50);
98. }
99. void verify() {
100. Timer1.detachInterrupt();
101. soilAverage = 0;
102. for (int i = 0; i < 100; i++) {
103. int soilHumidity = analogRead(SOIL\_HUMIDITY\_SIGNAL\_PIN);
104. soilAverage += soilHumidity;
105. Serial.print("SOIL: ");
106. Serial.println(soilHumidity);
107. }
108. Serial.println("SOILTOTAL: ");
109. Serial.println(soilAverage);
110. if ((soilAverage / 100) > 195) {
111. needWater = 1;
112. delay(100);
113. }
114. }
115. void dec2bin(int n) {
116. int c, k;
117. for (c = 15; c >= 0; c--) {
118. k = n >> c;
119. if (k & 1)
120. Serial.print("1");
121. else
122. Serial.print("0");
123. }
124. }
125. void dec2bin8(int n) {
126. int c, k;
127. for (c = 7; c >= 0; c--) {
128. k = n >> c;
129. if (k & 1)
130. Serial.print("1");
131. else
132. Serial.print("0");
133. }
134. }
135. void wait\_for\_dht11() {
136. delay(2000);
137. }
138. void start\_signal(uint8\_t dht11\_pin) {
139. pinMode(dht11\_pin, OUTPUT);
140. digitalWrite(dht11\_pin, LOW);
141. delay(18);
142. digitalWrite(dht11\_pin, HIGH);
143. pinMode(dht11\_pin, INPUT);
144. digitalWrite(dht11\_pin, HIGH);
145. }
146. void read\_dht11(uint8\_t dht11\_pin) {
147. uint16\_t rawHumidity = 0;
148. uint16\_t rawTemperature = 0;
149. uint8\_t checkSum = 0;
150. uint16\_t data = 0;
151. uint8\_t humi;
152. uint8\_t humd;
153. uint8\_t tempi;
154. uint8\_t tempd;
155. unsigned long startTime;
156. for (int8\_t i = -3; i < 80; i++) {
157. byte live;
158. startTime = micros();
159. do {
160. live = (unsigned long)(micros() - startTime);
161. if (live > 90) {
162. Serial.println("ERROR\_TIMEOUT");
163. return;
164. }
165. } while (digitalRead(dht11\_pin) == (i & 1) ? HIGH : LOW);
166. if (i >= 0 && (i & 1)) {
167. data <<= 1;
168. // TON of bit 0 is maximum 30 usecs and of bit 1 is at least 68 usecs.
169. if (live > 30) {
170. data |= 1;  // we got a one
171. }
172. }
173. switch (i) {
174. case 31:
175. rawHumidity = data;
176. break;
177. case 63:
178. rawTemperature = data;
179. case 79:
180. checkSum = data;
181. data = 0;
182. break;
183. }
184. }
185. Serial.println("Humidity: ");
186. dec2bin(rawHumidity);
187. Serial.print("\t");
188. humi = rawHumidity >> 8;
189. dec2bin8(humi);
190. Serial.print("\t");
191. rawHumidity = rawHumidity << 8;
192. humd = rawHumidity >> 8;
193. dec2bin8(humd);
194. Serial.print("\t");
195. Serial.print(humi);
196. Serial.print(".");
197. Serial.print(humd);
198. Serial.print("%");
199. Serial.println("");
200. Serial.println("Temperature Degree Celsius: ");
201. dec2bin(rawTemperature);
202. Serial.print("\t");
203. tempi = rawTemperature >> 8;
204. dec2bin8(tempi);
205. Serial.print("\t");
206. rawTemperature = rawTemperature << 8;
207. tempd = rawTemperature >> 8;
208. //tempd = (byte)rawTemperature;
209. dec2bin8(tempd);
210. Serial.print("\t");
211. Serial.print(tempi);
212. Serial.print(".");
213. Serial.print(tempd);
214. Serial.print("C");
215. Serial.println("");
216. Serial.println("Checksum Byte: ");
217. dec2bin8(checkSum);
218. Serial.println("");
219. dec2bin8(tempi + tempd + humi + humd);
220. Serial.println("");
221. if ((byte)checkSum == (byte)(tempi + tempd + humi + humd)) {
222. Serial.print("CHECKSUM\_OK");
223. } else {
224. Serial.print("CHECKSUM\_ERROR");
225. }
226. Serial.println("");
227. Serial.println("");
228. Serial.println("");
229. }