KOÇ UNIVERSITY

ELEC 339 / SENSORS CLASS
FALL 19

TERM PROJECT

SMART MONITORING SYSTEM FOR FARMING

Deniz KARADAYI
Kaan TOYAKSİ

16.12.2019

Project Description

The implemented project is a control system which can monitor and improve the conditions of temperature, humidity and light intensity for a given particular plant. The system is designed for small and private farming purposes which is held under unnatural light and temperature resources. It measures the temperature, light intensity and humidity of the interested plants, and gives warning in case that non-optimal conditions are detected for them. By doing this, the system aims at providing the perfect artificial conditions for those plants to raise.

Algorithm of the Project

Background Knowledge of Agriculture

At the end of our research on plant raising, we generally defined 6 types of category to specify the plant. The season that the plant prefers to be raised, and the water intake preference are the main separation points for us. Winter, summer and normal seasons, and water love and hate of the plant says us the optimal temperature, humidity and light intensity which are optimal for the interested plants. In addition, the optimal conditions also change according to the day and night times. Therefore, we wrote such an algorithm that can cover all possible cases and their effects on the plant's optimal preferences. According to the cases, we designed different warning mechanisms and status notifications.

Program Execution and Algorithm

At the beginning of the program, it is asked for user to specify the type of plants in terms of season and water preference. As a default, the considered time for the program is day time. It is stated that user can change it by pressing the button whenever he/she prefers to. After some information declarations related to the program, it goes into an infinite monitoring loop which consists of successive temperature, light intensity and humidity analyses respectively. For each factor, the program prints out the current value of the measurements, and comment on the status of the plant according to the parameters. Next, warns the user with different methods for different non-optimal cases. At the end of each loop, it prints out the general status of the plant, and makes suggestion on which factor should be checked.

If any button event occurs, which is handled by an interrupt routine, during the execution of a loop, program waits until the end of the loop to change the mode, Thusly, we prevent misleading outcomes from being stated to the user.

1. Temperature Effect and Warnings

No matter what season type the plant is, the optimal temperature decreases when it is night time. From the day time free point of view, the optimal temperature decreases from summer to winter season plants. Water preferences do not have a dramatic effect on optimal temperature. Therefore, by considering the all of these effects, we define 6 different temperature interval for plant types.

- $T < 7$: Cold for all types no matter what time it is.
- $7 < T < 13$: Optimal for winter plants at night, cold for others
- $13 < T < 18$: Optimal for winter plants at day and normal plants at night.
- $18 < T < 24$: Optimal for normal plants at day and summer plants at night.
- $24 < T < 30$: Optimal for summer plants at day, hot for others

- T > 30 : Hot for all types no matter what time it is.

As a warning, we generate 4 successive noises from the buzzer which are at different frequencies for hot and cold alarm.

2. Light Effect and Warnings

For night cases, all plant types hope for darkness. For day times, only winter plants prefer to be exposed comparatively less light. Therefore, the difference shows up for low light and high light cases. While winter plants do not like high intensity light exposure, non-winter plants do not like low light intensities at day time. Similarly, water preference does not have a dramatic effect on light preferences.

- Lux < 20 : Dark, only optimal at night mode
- 20 < lux < 110 : Optimal for winter plants, non-winter plants wish lighter cases.
- 110 < lux < 370 : Optimal for all plants at day mode
- Lux > 370 : Optimal for non-winter plants, winter plants wish darker cases.

As warnings, we use yellow LED blinks for second degree warnings which can happens at (20 < lux < 110) and (lux > 370) at day mode. For first degree warning, we use red LED blinks which are defined as light cases at night mode and dark case at day mode. To compensate the low light cases, we turn on the needed colours of an RGB LED as well, and turn off all colours for high light cases.

3. Humidity Effect and Warnings

Optimal humidity conditions only depend on the water preferences. Water-hater plants, which are not exactly haters, do not like high percentage of water intake. On the other hand, water-lover plants like higher percentage of water intake.

- hum% < 40 : Too dry, non-optimal for all types
- 40 < hum% < 50 : Optimal for water-haters
- 50 < hum% < 65 : Optimal for water-haters
- hum% > 65 : Too humid, non-optimal for all types

As warnings, we use turned on white LED to indicate dryness, and blinking white LED to indicate high humidity.

Main Components of the System

Arduino Uno R3 board is used for the project as project development environment, which contains ATmega328 microprocessor on it. It has digital and analog input-output pins, serial communication hardware and interfaces, external and timer interrupts, timer and chronometer properties

and so on [1]. As external sensors and devices, an LM35 temperature sensor, an LDR photo-resistor, a humidity sensor, a HC05 Bluetooth module are used as inputs. A buzzer, a button, a RGB LED, and 3 LEDs which are red, yellow and white are used as outputs.

INPUTS	**I/O PINS**	**PROPERTY**
LM35 – Temperature Sensor	A0 - tempPin-	Measures the temperature of the environment by its varying resistance.
Photo-resistor	A5 - lightPin	Measures the light intensity of the environment by its varying resistance.
Gravity: Analog Soil Moisture Sensor	A1 - humPin	Measures the humidity of the environment by its varying capacitance.
HC05 – Bluetooth Module	0 – RX 1 - TX	Provides the wireless connection between PC and Arduino.
Button	2	Shifts between day and night
OUTPUTS	**I/O PINS**	**PROPERTY**
Buzzer	9 - buzzer	Alarm Mechanism for Temperature: Generates sound at different frequencies and rates according to the value of the temperature.
LED Yellow	10 – yellowPin	Alarm Mechanism for Light: Blinks when light intensity is low.
LED Red	8 - redPin	Alarm Mechanism for Light: Blinks when light intensity is too low.
LED RGB - Red	6 - R	Compensates the light by turning on when the light intensity is low.
LED RGB - Green	4 – G	Compensates the light by turning on when the light intensity is low.
LED RGB - Blue	5 - B	Compensates the light by turning on when the light intensity is too low.
LED White	3 – whitePin	Alarm Mechanism for humidity: Blinks at different rates when humidity is not optimal..

- ***LM35:***

This sensor works by thermo-resistive principles. Its internal resistances changes according to temperature. It has an already calibrated circuit in terms of Celsius which is linear function of the voltage. The transfer function of the analog output of the LM05 is given as following. [9]

$$Vout = \frac{10\,mV}{°C} * T$$ where T is temperature in Celsius. [9]

Its operating voltage is from 4V to 30V, the minimum temperature change it can recognize is 0.5 °C at 25°C. The ADC on the Arduino has 10 bits resolution which is mapped to 5V [9]. Therefore, the minimum voltage change it can recognise is 5000mV / 2^{10}bits = 4.89 mV. It corresponds to 0.49 °C, which

also nearly equals to the physical noise level of the sensor. Its self-heating factor is approximately 0.08°C, which clearly ignorable when considering the sensitivity.

In brief, it was pretty trouble-free option for us to measure temperature. In addition, it is also low-cost and small sensor, also very compatible with the Arduino development environment. Therefore, we prefer to use it in our project.

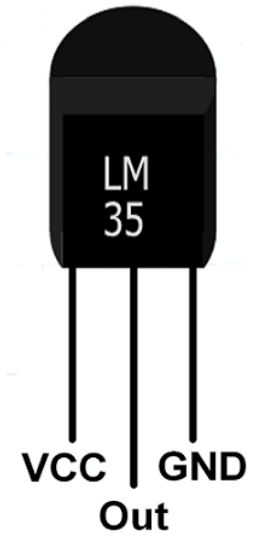


Figure 1: LM35 Illustration with Pinouts [8]

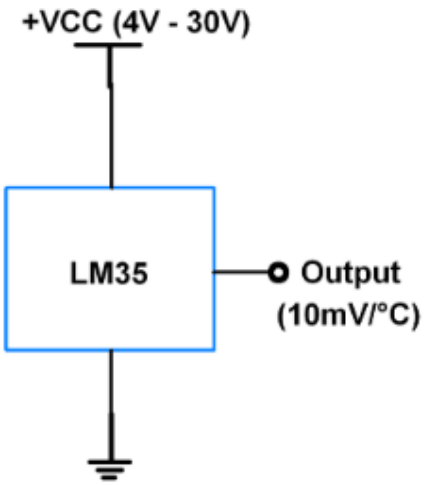


Figure 2: Basic Circuit setup to measure temperature with LM35 [8]

- ***LDR***

LDR means Light Dependent Resistor which it is easy to understand its working principle by the name. The resistance of the resistor decreases exponentially as the light intensity decreases. There is no additional internal circuit inside; thus, it is very cheap and flexible to use.

The typical behaviour of the sensor is larger than 1 MΩ at dark, and smaller than 1 kΩ at high light levels [7]. The response time vary from 2 to 50 ms at normal circumstance, but it takes roughly 10 secs after a direct removal of a light source which is not very good response time in general [7]. However, in our project, the instant reaction is not required, so it is acceptable for the project cases.

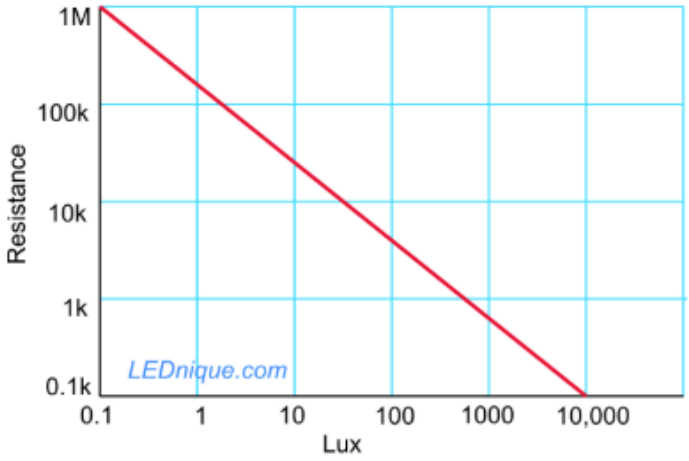


Figure 3: Resistance (?) Change Graph with respect to Light Intensity (lux) [7]

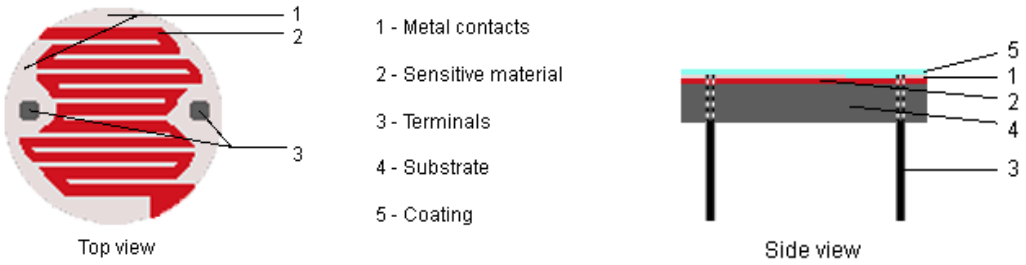


Figure 4: Top and Side Views of LDR [2]

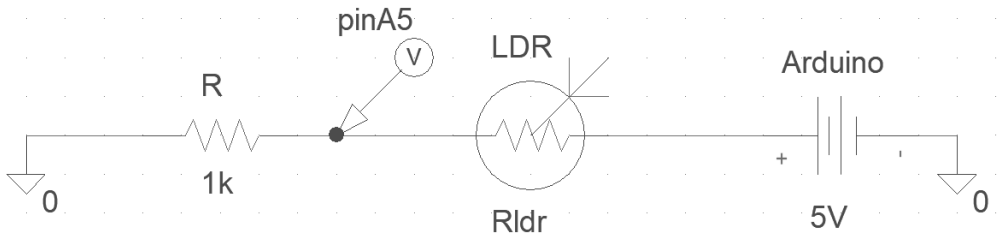
To use LDR value appropriately, we need a constant resistor. In our case, we chose 1 kΩ resistor. To obtain the circuit we design the setup in the Figure 4. Supply 5V from Arduino to LDR and connect it to the resistor. Next, we connected to the analog pin A5 to between them such that the reading will correspond to the voltage of the constant resistor. The initial thing we need to find is resistance of LDR, which can be found by the following calculations.

$$V_{ldr} = 5V - \frac{(pinA5 * 5V)}{1023}$$

$$R_{ldr} = \frac{V_{ldr}}{I} = \frac{V_{ldr}}{\frac{V_r}{R}}$$

The relation between R_{ldr} and intensity of light was not linear. In order to calibrate it, we need data of resistances at different intensities and plot them. The best fitted graph of the plot will approximately give us the transfer function, which is in our case is the following. [1]

$$Light\ Intensity\ (lux) = \frac{R_{ldr}^{\frac{1}{-0.8616}}}{10^{\frac{5.118}{-0.8616}}} \qquad [5]$$



- ***Gravity: Analog Soil Moisture Sensor for Arduino***

The humidity sensor we have selected is a kind of cheap a primitive option. However, it is pretty enough for small farming purposes like we consider. It has two parallel and separated probes, which will be stuck into the soil. Between these probes, current would be able to pass when there is a conductor between them. Therefore, the much amount of water between the probes exists, the much easily the current pass through them [3].

The supply voltage for it is 3.3V to 5V, and its output voltage varies between 0 to 4.2V [3]. The mechanism inside of the sensor work such that the output voltage increases as the humidity decreases. Therefore, we read 1023 from Arduino as analog reading when there is no water, and approximately 350 when it is fully in water. In order to calculate the humidity in terms of percentage, we get mapped the interval 1023 – 350 to the interval 0 – 100 %.

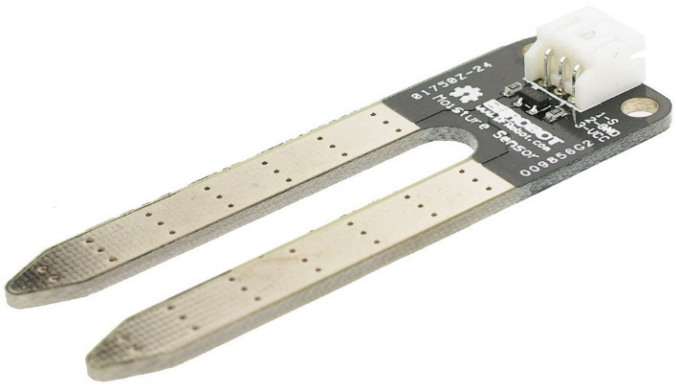


Figure 6: Analog Moisture Sensor [3]

- ***HC05 Bluetooth Module***

HC05 is an easy-use Bluetooth apparatus to prefer as wireless network provider for small distant projects. It is a serial port protocol device with 3 Mbits per second transmission rate and 2.4 GHz baseband [6]. The typical sensitivity of the module is -80dBm, and transmit power is up to 4dBm [6]. General operation voltage is 1.8 to 3.6V for input and output pins [6]. The Arduino supply 5V from its output pins, so it is wise to use voltage divider for input pins of HC05. It is available to use of UART interface with computer at proper baud rates, and HC05 has a pretty large range of supported baud rate options [6].

To connect HC05 to Arduino board, we need to provide the serial communication line between them which are RX and TX. RX of HC05 should be connected to TX of Arduino, and vice versa. The voltage divider is only necessary for the RX pin of the HC05 since it is the only input pin which receives data from Arduino.

The module can be configured as master or slave, and configuration can be done with universal AT commands. According to the blink types of the module, it is possible to understand the state of it which are possibly connected, waiting in data mode, and in command mode. Pin specifications can be seen at the following table [6].

Pin Number	Pin Name	Description
1	Enable / Key	This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default it is in Data mode
2	Vcc	Powers the module. Connect to +5V Supply voltage
3	Ground	Ground pin of module, connect to system ground.
4	TX – Transmitter	Transmits Serial Data. Everything received via Bluetooth will be given out by this pin as serial data.
5	RX – Receiver	Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth

6	State	The state pin is connected to on board LED, it can be used as a feedback to check if Bluetooth is working properly.
7	LED	Indicates the status of Module • Blink once in 2 sec: Module has entered Command Mode • Repeated Blinking: Waiting for connection in Data Mode • Blink twice in 1 sec: Connection successful in Data Mode
8	Button	Used to control the Key/Enable pin to toggle between Data and command Mode

Figure 7: Pin Configurations of HC05 Bluetooth Module [4]

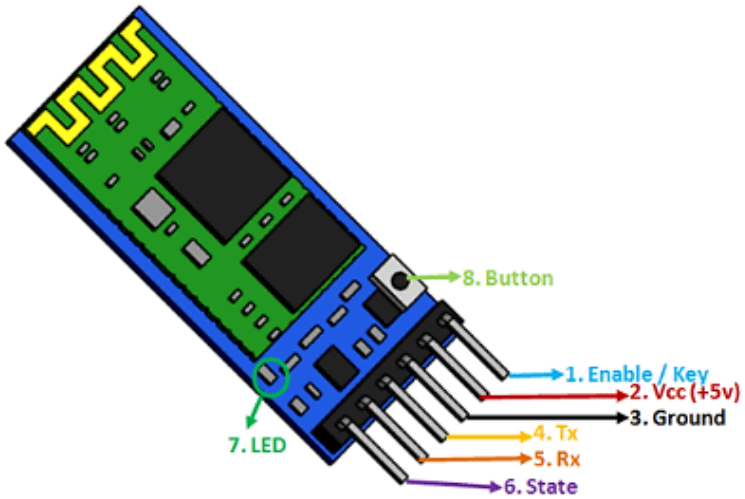


Figure 8: Illustration of HC05 Bluetooth Module [4]

References

1. Arduino , "Arduino UNO Datasheet," *Octopart*. [Online]. Available: https://datasheet.octopart.com/A000066-Arduino-datasheet-38879526.pdf.

2. H. Melih. Erdoğan, "LDR," *Technology Laboratory*. [Online]. Available: http://technologylaboratory.blogspot.com/p/ldr.html. [Accessed: 18-Dec-2019].

3. "Gravity: Analog Soil Moisture Sensor For Arduino," *direnc.net*. [Online]. Available: https://pdf.direnc.net/upload/gravity-analog-toprak-nem-sensoru-datasheet.pdf.

4. "HC-05 Bluetooth Module Pinout, Specifications, Default Settings, Replacements & Datasheet," *Components 101*, 10-Mar-2018. [Online]. Available: https://components101.com/wireless/hc-05-bluetooth-module. [Accessed: 18-Dec-2019].

5. Instructables, “Measuring Light Using Light Sensor,” *Instructables*, 29-Sep-2017. [Online]. Available: https://www.instructables.com/id/Measuring-Light-Using-Light-Sensor/. [Accessed: 18-Dec-2019].

6. ITead Studio, “HC-05 -Bluetooth to Serial Port Module ,” *Electronicae Studio*, 18-Jun-2010. [Online]. Available: http://www.electronicaestudio.com/docs/istd016A.pdf.

7. “Light dependent resistor (LDR),” *LEDnique*. [Online]. Available: http://lednique.com/opto-isolators-2/light-dependent-resistor-ldr/. [Accessed: 18-Dec-2019].

8. “LM35 Temperature Sensor: Sensors & Modules,” *ElectronicWings*. [Online]. Available: https://www.electronicwings.com/sensors-modules/lm35-temperature-sensor. [Accessed: 18-Dec-2019].

9. Texas Instruments, “LM35 Precision Centigrade Temperature Sensors” LM35 datasheet, Aug. 1999 [Revised Dec. 2017].