Supervised by:

Mr Ravi Luthra

Mr Naveen Sharma

By

Shivam Singha

Sanskar Chakraborty

Rankan Sen

Kendriya Vidyalaya, New Bongaigaon

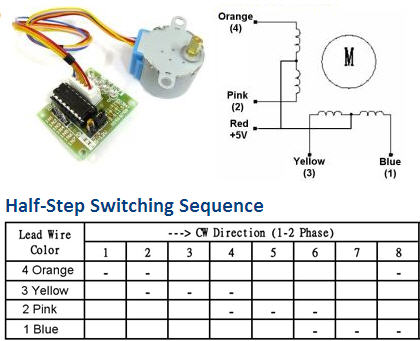
ASSAM

Auto irrigation system

Jawaharlal Nehru NSME Exhibition 2018

**Hardware**

1. **Arduino Uno:** We used Arduino Uno, a microcontroller prototype board, to sense the signal from the soil moisture sensor and release the valve to dispense the water from the reservoir. The automation is done by flashing a compiled C++ code using the Arduino IDE.
2. **Stepper Motor:** We used 28BYH-48 Stepper Motor with ULN2003A motor driver to release the valve. Since Stepper motor produces a lot of torque, it is able to open the tightly closed valve. The motor has 4 coils of wire that are powered in a sequence to make the magnetic motor shaft spin. When using the full-step method, 2 of the 4 coils are powered at each step. The 28BYH-48 datasheet specifies that the preferred method for driving this stepper is using the half-step method, where we first power coil 1 only, then coil 1 and 2 together, then coil 2 only and so on…With 4 coils, this means 8 different signals, like in the table below. Since, the built-in Arduino Stepper library do not support half-step method. We used the <AccelStepper.h> library, which is further discussed in the ‘Software’ section.



**Half-Step Switching Sequence**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lead Wire Color | Clockwise Direction | | | | | | | |
| Coil 1 | Coil 2 | Coil 3 | Coil 4 | Coil 5 | Coil 6 | Coil 7 | Coil 8 |
| 4 – Orange | **✓** | **✓** |  |  |  |  |  |  |
| 3 – Yellow |  | **✓** | **✓** | **✓** |  |  |  |  |
| 2 – Pink |  |  |  | **✓** | **✓** | **✓** |  |  |
| 1 – Blue |  |  |  |  |  | **✓** | **✓** | **✓** |

1. **Stepper Motor Driver:** The ULN2003A is an array of seven NPN Darlington transistors capable of 500 mA, 50 V output. It features common-cathode flyback diodes for switching inductive loads. The ULN2003 is known for its high-current, high-voltage capacity. The drivers can be paralleled for even higher current output. Even further, stacking one chip on top of another, both electrically and physically, has been done. In our system, it is used for interfacing with a stepper motor, where the motor requires high current ratings which cannot be provided by other interfacing devices.
2. **LCD Display:** We used a 16 characters - 2 lines LCD display to indicate the status of water flow and soil moisture percentage. The LCD Display is illuminated by backlight and is well visible at both day and night, perfect for garden use. JHD162A is the LCD module used here. JHD162A is based on the HD44780 driver from Hitachi. The JHD162A has 16 pins and can be operated in 4-bit mode (using only 4 data lines) or 8-bit mode (using all 8 data lines). Here we are using the LCD module in 4-bit mode. The control and data pins are directly connected to Arduino UNO’s Digital pins. A 10kΩ potentiometer is connected to the wiper pin which is used to adjust the contrast of the LCD Display. An optional 5V and Ground with a 220Ω resistor is connecter to the backlight pin or the blue illumination.

**Software**

We used C++ programming language to automate the process of sensing and dispensing water.

For Stepper Motor, the <AccelStepper.h> library was used. The <AccelStepper.h> library supports the Half-Step method which is recommended by the manufacturer of the stepper motor as discussed in the ‘Hardware’ section. AccelStepper significantly improves on the standard Arduino Stepper library in several ways: It supports acceleration and deceleration. It supports multiple simultaneous steppers, with independent concurrent stepping on each stepper. API functions never delay() or block. It supports 2, 3 and 4 wire steppers, plus 3 and 4 wire half steppers Very slow speeds are supported. This code uses speed calculations as described in "Generate stepper-motor speed profiles in real time" by David Austin with the exception that AccelStepper uses steps per second rather than radians per second (because we don’t know the step angle of the motor) An initial step interval is calculated for the first step, based on the desired acceleration. On subsequent steps, shorter step intervals are calculated based on the previous step until max speed is achieved.

In the programming part, to facilitate communication between Arduino and LCD module, we make use of a built-in library in Arduino <LiquidCrystal.h> – which is written for LCD modules making use of the Hitachi HD44780 chipset. This library can handle both 4-bit mode and 8-bit mode wiring of LCD. In 4-bit mode, data is sent using 4 data pins and 3 control pins. In our project, R/W pin is always grounded so we require only 6 pins in 4-bit mode, thus saving no of pins. During interfacing the library is first initialized and then define pins using the command LiquidCrystal lcd(RS, E, D4, D5, D6, D7), pins are assigned in this order.

The code can be further extended to regulate the flow of water for individual plants, fulfilling requirements of different plants adequately without human interference.

**Working**

The soil sensor is basically two galvanized iron nails each 2 inches long. 5V is supplied to one of the nails and the other nail is connected to the A0 Analog Input point on Arduino UNO. analogRead() method is used to sense the current flowing in through the analog input pin. If soil contained water, the resistance decreases and more current flows through the nails, and thus the analogRead() returns a higher value. We have set a threshold current under which the soil is considered dry. Once the signal goes under that threshold, the stepper motor connected with the driver is run. The stepper motor unwinds the valve (tap) and releases the water from the reservoir. The valve stays open for a set interval of time and closes after that time. After closing the valve, Arduino starts sensing the signals from soil sensor again. If the soil moisture goes under the same threshold, the above-mentioned process is repeated again. As the water flow is controlled, the LCD keeps displaying the status.

**Uses**

Our Auto irrigation system could be modified only a bit to implement it in large fields. It could be configured to supply different amounts of water to irrigate each plant according to its requirements. The large-scale implementation of this system only requires using a large water pump in place of the stepper motor actuated valve (or tap). Manual control of the water flow is also available through the included buttons to manually turn the water flow on or off in case the automatic system wrongly irrigated.