Flood gate control system

Curriculum for Applied Learning

Digital Logic and Design – CSE1003

16BEC0403	SANSKAR BISWAL
16BCE2221	NEHA BINDAL

Dr Saritha V
Associate Professor

B. Tech.
In
Computer Science and Engineering

School of Computer Science & Engineering

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)

Abstract

The aim of this project was to make a floodgate control system to automate the entire process of flood-gate control and water-release systems in dams. The focus of this project was to use an Arduino to control a stepper motor which will move the gate at a certain angle depending on the level of water . The topic seeks to address the challenge of reducing or even removing human-induced errors in the process. The inspiration of the project comes from a recent incident where the flood-gates of a dam were opened erroneously which resulted in a lot of human casualties. The project is thus an attempt to address such issues.

Table of Contents:

1. Introduction

1.1 Problem Statement

The topic seeks to address the challenge of reducing or even removing human-induced errors in the process. The inspiration of the project comes from a recent incident where the flood-gates of a dam were opened erroneously which resulted in a lot of human casualties. The project is thus an attempt to address such issues.

2. Project Plan

2.1 Components Required

- Arduino UNO
- stepper motor
- ultra sonic sensor
- bread board
- jumper wires
- UN2004 (Darlington array)
- Potentiometer

2.1.1 Software Requirements

A software is needed to be able to program and test the Arduino board being used in the project. The software used to do so is the Arduino IDE available for computers.

//2.1.2 Hardware Requirements

The project emulates the setting of a room with two doors. The components used for achieving this are –

- Arduino UNO microcontroller - 1
- Arduino power cable – 1
- Power bank - 1
- Breadboard – 1
- Ultra Sonic sensors – 2
- Jumper wires

2.1.3 Description of Components

- Ultra sonic sensors – Used to detect water level
- Arduino UNO microcontroller – The brain of the flood gate system. Used to interpret signals from the Ultrasonic sensor and control the stepper motor.
- Breadboard – The base of the circuit assembly. All connection made onto the breadboard.
- Jumper wires – Connections between various modules made using male to male jumper wires.
- Arduino power cable – Connect the Arduino to the power bank to provide for the power requirements of the circuit

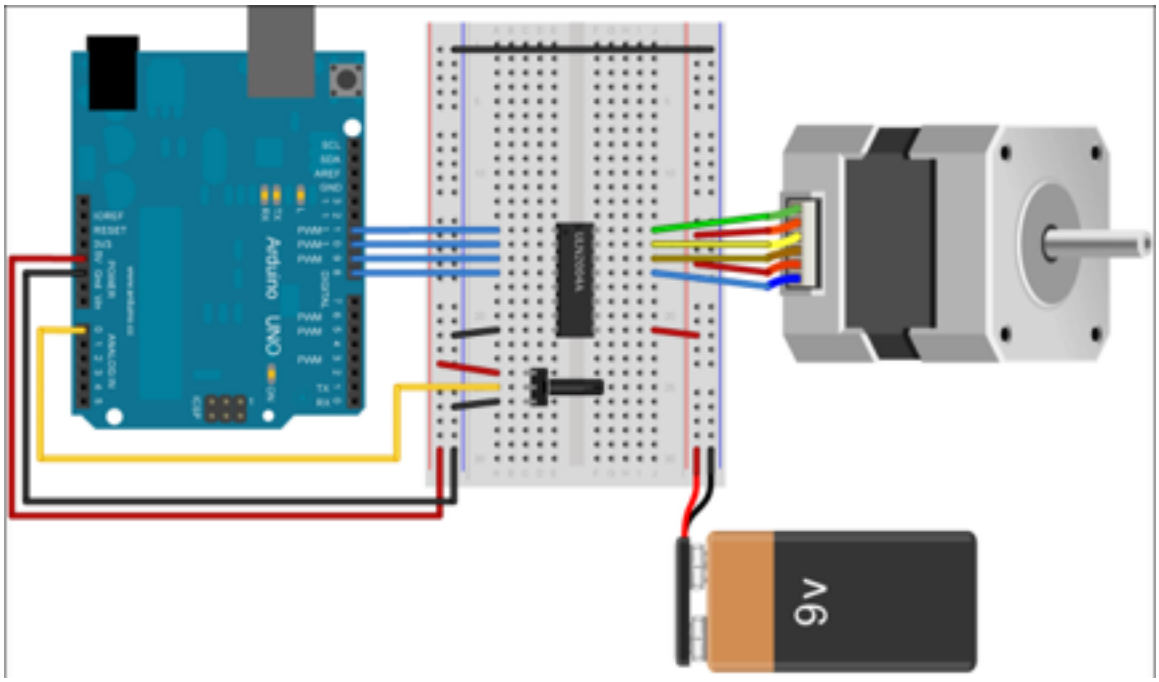
2.2 Cost analysis

- Arduino UNO – 600 Rupees
- Breadboard – 70 Rupees
- Ultra Sonic sensors – 180 Rupees
- LEDs – 10 Rupees each
- Jumper wires – 5 Rupees each
- Motor- 220 Rupees

Total project cost –1085 Rupees.

3. Design

3.1 Circuit Diagram



3.1 Circuit Explanation:

The circuit is divided into three major sections –

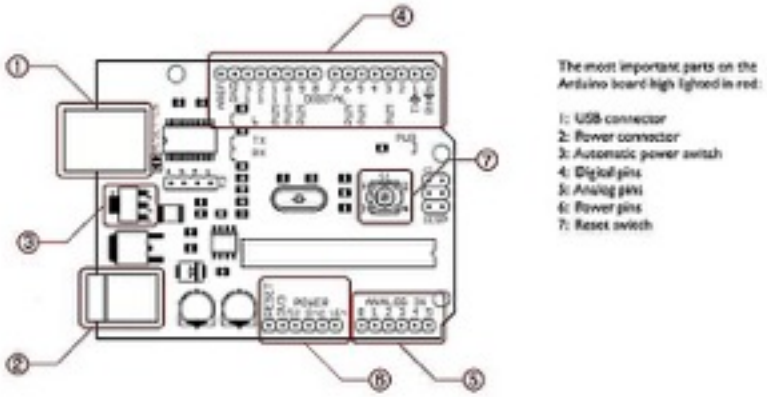
- Sensor section –The Ultrasonic sensor determines the water level of the reservoir
- Control section – The Arduino UNO controls the whole process of the project. The signals from the sensor are interpreted to change the state of the gate and the corresponding output is given via Unipolar Stepper motor.
- Motor section : The motor is controlled by a 4 pin grade array called Darlington array. This controls the no.of rotations and revolutions that the motor should make to ensure that the dam flood gates maintain optimum level.

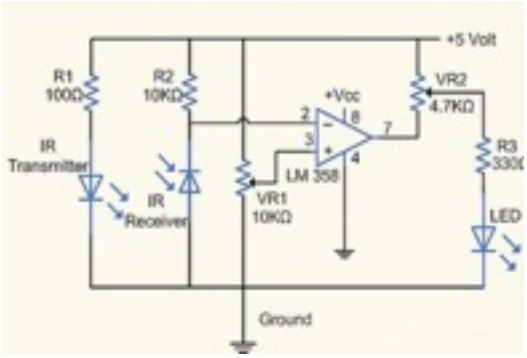
3.2 Modules

The modules used along with the Arduino microprocessor are the LCD display, the IR sensor and the LEDs.

3.2.1 Module wise circuit diagram and its description

➢ Arduino UNO - Arduino/Genuino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button.





3.3 Description:

The Sensor works on the principle of SONAR. It emits frequency of 40000Hz. The echo captures this and by the lapse in time, determines the distance travelled. This is equivalent to the distance from overflow. This enables us to maintain a constant monitor on the water level of the said reservoir.

The stepper motor is controlled in steps of its motor winding usually in a 360 degree fashion.

3.4 Working:

- ➢ The ultrasonic sensor detects the change in the level of water in a reservoir.
- ➢ If this is within standard limits, then no action is taken.
- ➢ Otherwise, the motor is guided to send out an alarm and release water from the flood gates in a controlled fashion. Once the water is back to its stable level, the flood gate closes on its own.
- ➢ This was, the gate releases only that much as is needed and by extensions minimises losses due to flooding.

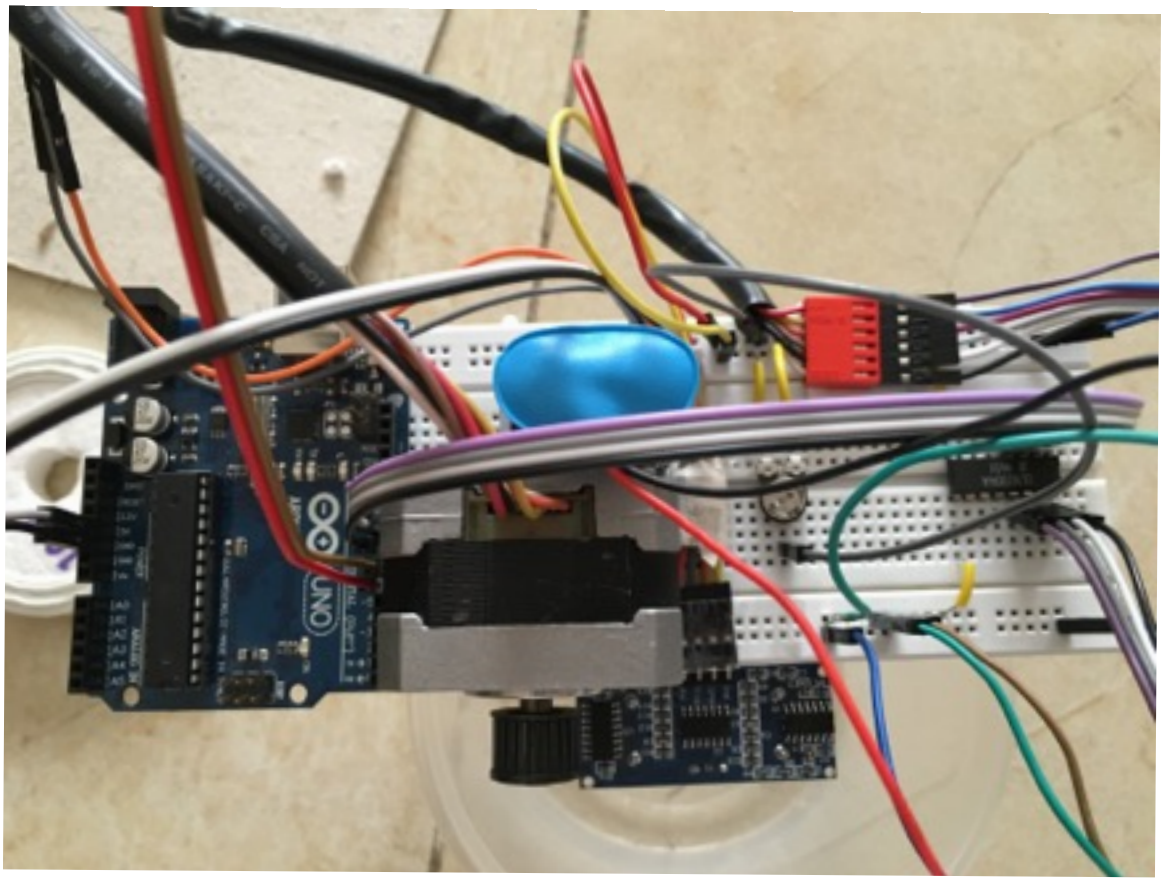
3.5 The Arduino code:

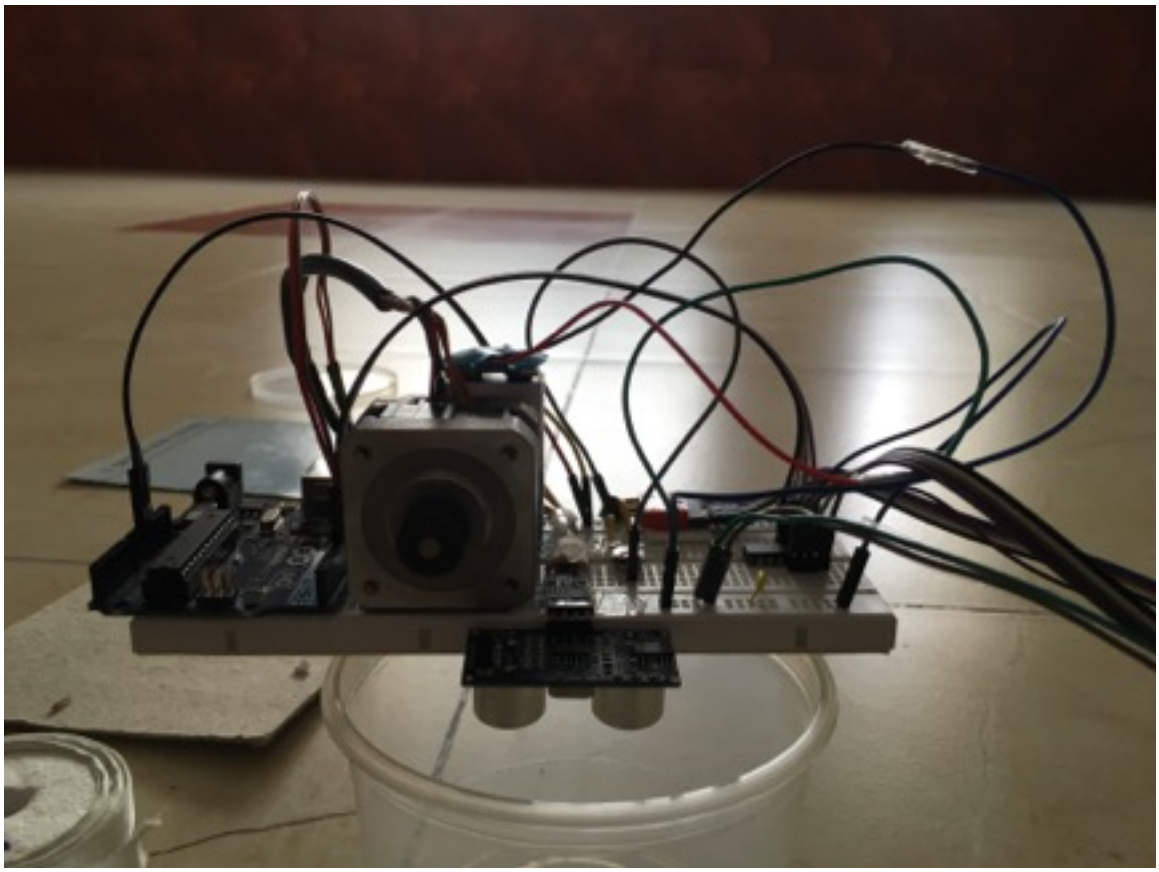
```
/*
 Stepper Motor Controller
 language: Wiring/Arduino

 This program drives a unipolar or bipolar stepper motor.
 The motor is attached to digital pins 8 and 9 of the Arduino.

 The motor moves 100 steps in one direction, then 100 in the other.

 Created 11 Mar. 2007
 Modified 7 Apr. 2007
 by Tom Igoe

 */

// define the pins that the motor is attached to. You can use
// any digital I/O pins.

#include <Stepper.h>

#define motorSteps 200     // change this depending on the number of steps
                           // per revolution of your motor
#define motorPin1 8
#define motorPin2 9
#define motorPin3 10
#define motorPin4 11
#define ledPin 13

const int trigPin = 7;
const int echoPin = 6;

long duration;
int distance;
int d;

// initialize of the Stepper library:
Stepper myStepper(motorSteps, motorPin1,motorPin2,motorPin3, motor-
Pin4);
```

```
void setup() {
  // set the motor speed at 60 RPMS:
  myStepper.setSpeed(60);

  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input

  // Initialize the Serial port:
  Serial.begin(9600);

  //Set initial distance as 0
  d=0;
  // set up the LED pin:
  pinMode(ledPin, OUTPUT);
  // blink the LED:
  myStepper.step(100);
  blink(3);
}

void loop() {

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  // Sets the trigPin on HIGH state for 10 micro seconds
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Reads the echoPin, returns the sound wave travel time in microseconds
  duration = pulseIn(echoPin, HIGH);
  // Calculating the distance
  distance= duration*0.034/2;
  if(distance!=d){
    d=distance;
    Serial.println(d);
    if(d>20){
      Serial.println("Water Level are within limits");
      delay(1000);
    }else if(d<=20 && d>10){
      Serial.println("Water Levels on the rise. Precaution Necessary");
```

```
      blink(20);
      delay(1000);
    }else if(d<=10 && d>5){
      Serial.println("Level critical. Releasing water");
      blink(50);
      myStepper.step(-30);
      delay(1000);
    }else{
      Serial.println("Overflow Imminent. Opening All floodgates");
      myStepper.step(100);
      delay(1000);
      myStepper.step(-100);
      delay(1000);
    }
  }else{
    Serial.println("No change in water level");
    delay(1000);
  }
}

// Blink the reset LED:
void blink(int howManyTimes) {
  int i;
  for (i=0; i< howManyTimes; i++) {
    digitalWrite(ledPin, HIGH);
    delay(200);
    digitalWrite(ledPin, LOW);
    delay(200);
  }
}
```

3.6Snapshots:





4. Advantages:

- ➢ removes manual intervention
- ➢ low cost
- ➢ high efficiency

Disadvantages:

- ➢ It's possible to fool the system by artificially placing obstacles

5.Applications:

- ➢ Automated control of flood gates.

Future Expansion:

By applying this circuit we can prevent manual intervention of flood gates and thus prevent casualties due to unplanned release of reservoir water.

6. Conclusion

By taking power from the battery, the project performs all the functions that were planned to be executed.

The Ultra sonic sensors detect the rise and fall in the level of water and transfer the signals to the Arduino board. The Arduino board then suitably controls the movement of the stepper motor.

The stepper motor thus controls the movement of the flood gate.

All in all, the project is able to create a prototype of automated flood gates which was successfully achieved.

7. References

➢ www.arduino.cc

➢ www.youtube.com