CSE 316 Project Report
Automatic Floodgate Controller
Using ATmega32 Microcontroller

Ali Haisam Muhammad Rafid
Student ID: 1405013
Md. Toufikuzzaman
Student ID: 1405015

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology

January 17, 2018

Contents

1 Introduction

The purpose of this project is to automatically control river dam floodgate depending on water level. If water level rises gate will be opened gradually to keep water flow in control. If the water level is very high, an alarm will turn on. Similarly if water level decreases gate will be closed gradually so that enough water can be stored. Though our main focus was to automate gate actions there was manual override button and two other buttons to open and close the gate manually. We displayed the readings on 16x2 LCD Display and used Sonar Sensor to measure height. Also DC Motor and Motor Driver was used to control the gate.

2 Hardwares Used

List of hardwares used and their price:

Hardware Name	**Unit(s)**	**Price(Taka)/ Unit**	**Total Price(Taka)**
ATmega32	1	150	150
AVR ISP	1	200	200
M-M Jumper Wire	2 lot	70	140
M-F Jumper Wire	2 lot	70	140
16x2 LCD Display	1	130	130
IC Rail	1	7	7
Bread Board Regular	2	80	160
Bread Board Medium	2	75	150
DPST Switch	3	4	12
Buzzer	1	25	25
Sonar (HC-SR04) Sensor	1	110	110
DC Motor (Gear-less)	1	50	50
Motor Driver (LN-298N)	1	210	210
9V Battery	1	35	35
Battery Clipper	1	30	30
			Total Cost: 1549

Initially, we bought most of the components from Patuatuli, but while doing our project, we bought Motor Driver, extra 9V Batteries, DC Motor, Medium Sized Breadboard and Jumper Wires from an online shop called **Gyanjam** which delivers in BUET.

3 Softwares Used

- **Atmel Studio 6.2:** For compiling .c file and generating .hex file.
- **ProgISP:** For writing the .hex file into ATmega32.

- **Proteus 8 Professional:** For drawing circuit diagram and simulating the circuit.

4 Flow Chart

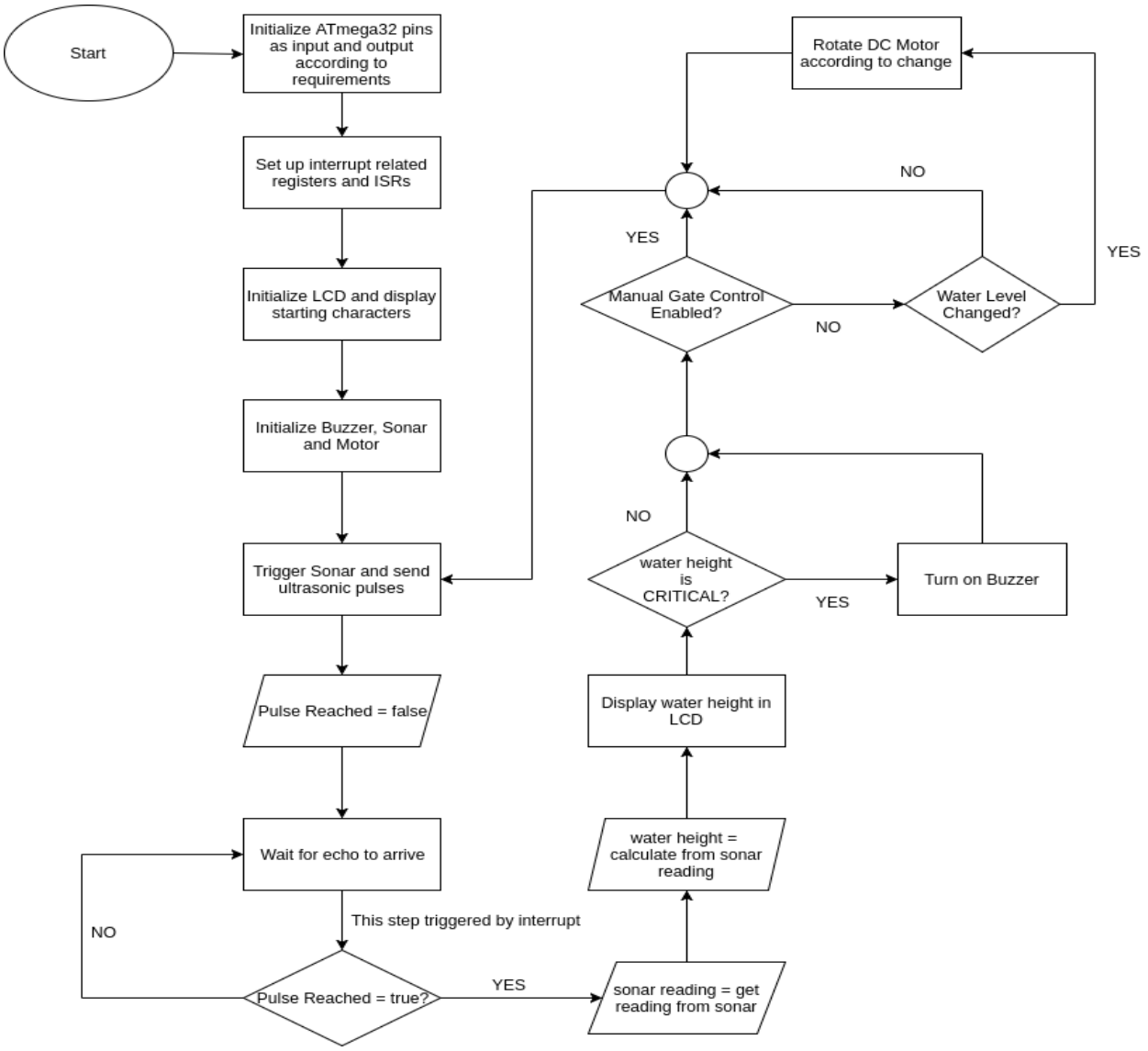

Figure 1: Flow Chart for Automatic Gate Control

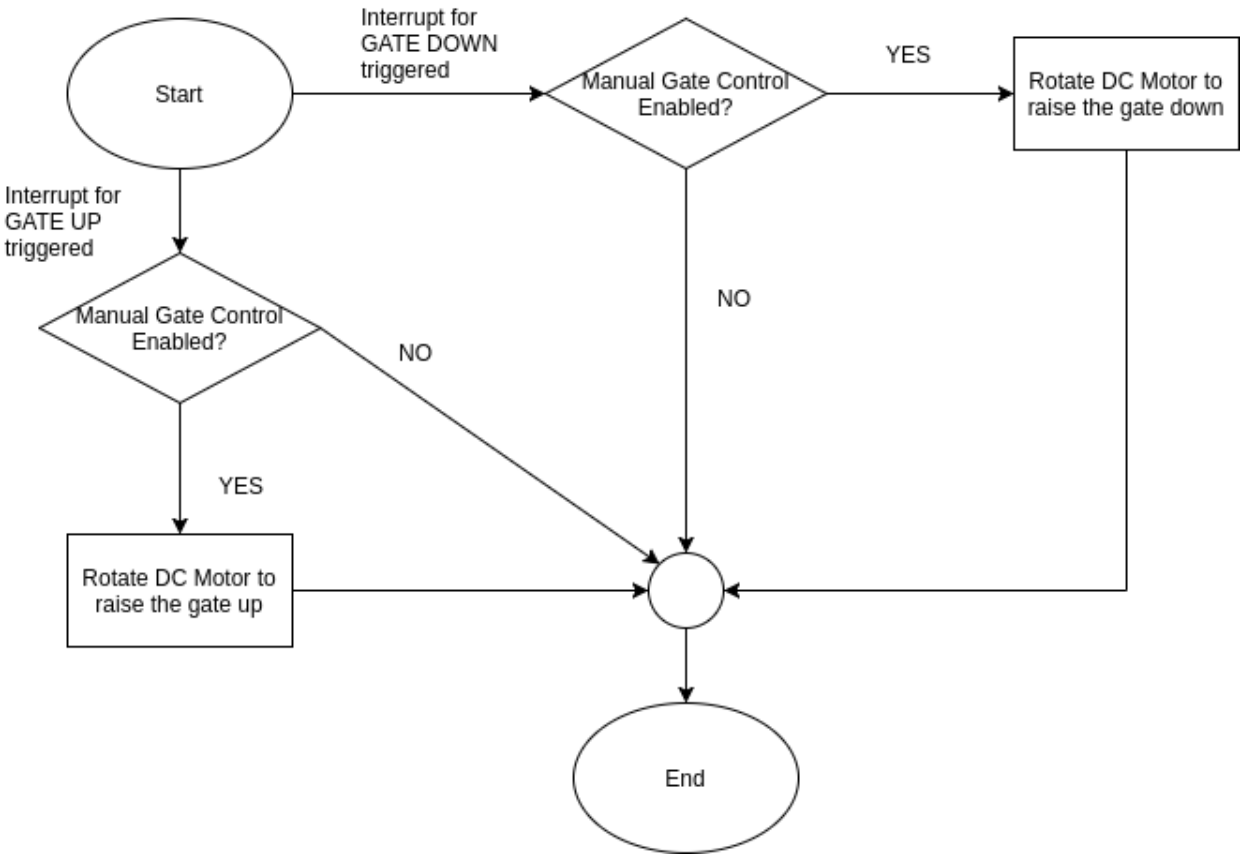

Figure 2: Flow Chart for Manual Gate Control

5 Block Diagram

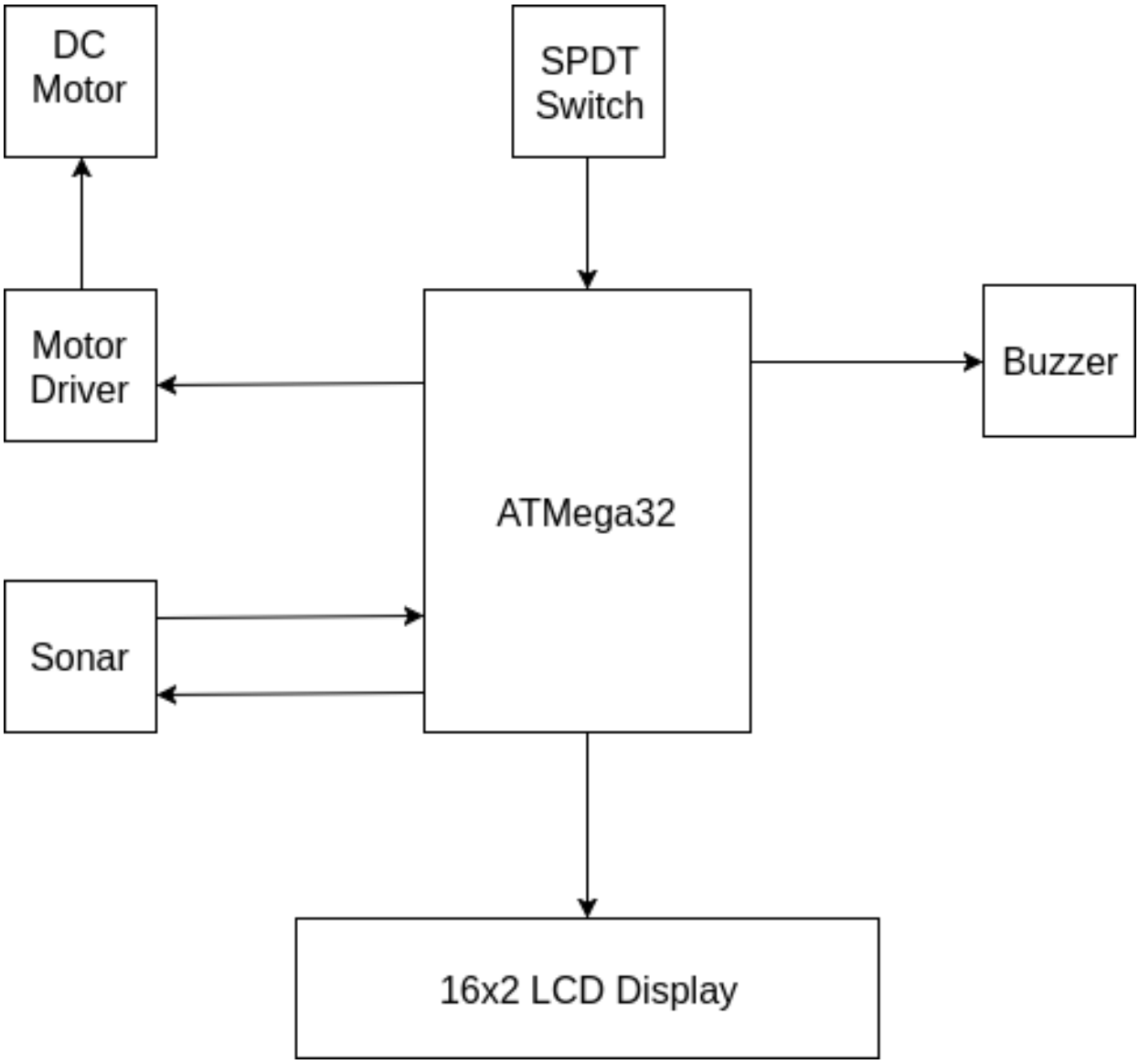

Figure 3: Block Diagram Showing Input and Output

6 Circuit Diagram

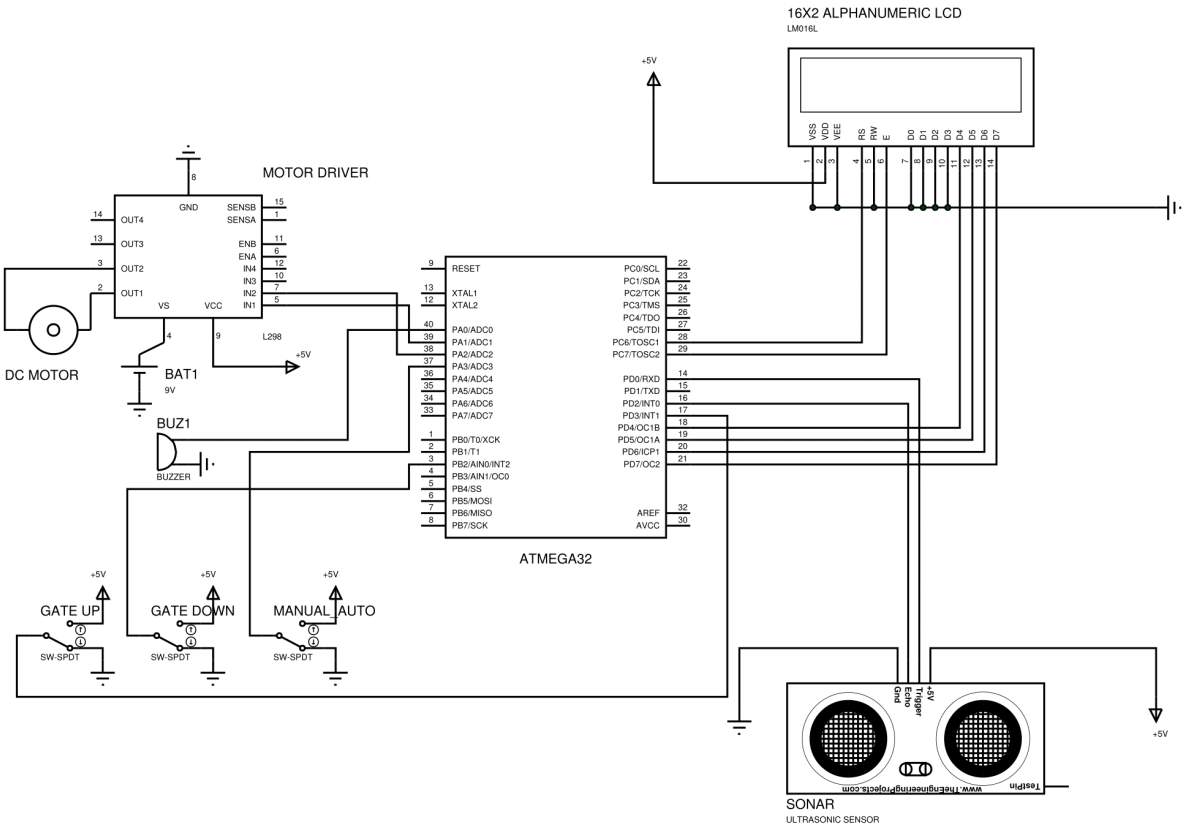

Figure 4: Circuit Diagram

7 Basic Working Principle

7.1 16x2 LCD Display

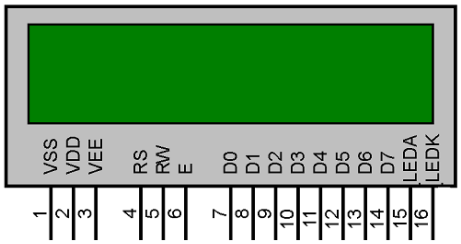

Figure 5: 16x2 LCD Display

16x2 LCD display has 16 pins. We used 4-bit Data mode and D4-D7 pins were used as data pins. Functions for using LCD display were written in lcd.h header file. Connection of 16 pins are listed below:

- **LCD pin1 (VSS):** Connected to GND.
- **LCD pin2 (VDD):** Connected to VCC.
- **LCD pin3 (VEE):** Connected to GND to get maximum contrast.
- **LCD pin4 (RS):** Connected to pin C6(28) of ATmega32.
- **LCD pin5 (RW):** Connected to GND.
- **LCD pin6 (E)nable:** Connected to pin C7(29) of ATmega32.
- **LCD pin7 (D0):** Connected to GND.
- **LCD pin8 (D1):** Connected to GND.
- **LCD pin9 (D2):** Connected to GND.
- **LCD pin10 (D3):** Connected to GND.
- **LCD pin11 (D4):** Connected to pin D4(18) of ATmega32.
- **LCD pin12 (D5):** Connected to pin D5(19) of ATmega32.
- **LCD pin13 (D6):** Connected to pin D6(20) of ATmega32.
- **LCD pin14 (D7):** Connected to pin D7(21) of ATmega32.
- **LCD pin15 (LEDA):** Connected to VCC. Used to lit backlight.
- **LCD pin16 (LEDK):** Connected to GND. Used to lit backlight.

LCD Module was used to display the Sonar Reading and Water Level.

7.2 Sonar (HC-SR04) Sensor

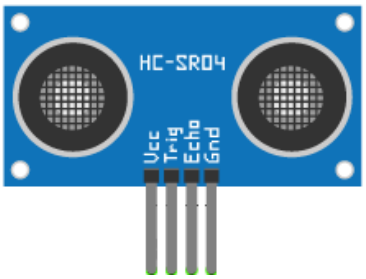

Figure 6: Sonar (HC-SR04) Sensor

Sonar sensor has 4 pins. Trigger pin triggers the sensor to generate ultrasonic wave and echo pin receives the wave reflected of an object or in this case water surface. Echo pin triggers the ISR for interrupt 0 (INT0) when it starts waiting and finish receiving reflected wave. Time interval is calculated using Timer 0 of ATmega32 and using this value distance from water surface to sonar is calculated and hence the water height is determined. Connection of 4 pins are listed below:

- **VCC:** Connected to VCC.
- **GND:** Connected to GND.
- **Trigger:** Connected to pin D0(14) of ATmega32.
- **ECHO:** Connected to pin D2(16), INT0 pin of ATmega32.

7.3 Buzzer

Figure 7: Buzzer

In buzzer connection +ve end was connected to pin A0(40) of ATmega32 and -ve end was connected to GND. Buzzer was used to sound alarm when water level reaches a danger point.

7.4 Motor Driver and DC Motor

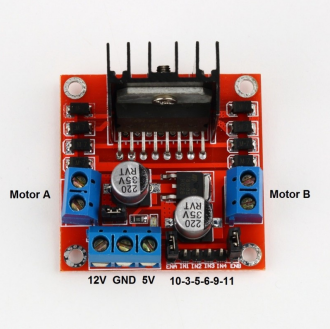

Figure 8: LN-298N Motor Driver

Figure 9: DC Motor

Figure 10: 9V Battery

DC Motor requires more current than ATmega32 can supply so we connected our motor with LN-298N Motor driver. This driver can supply power to two motors simultaneously. We used pins for Motor A. Connection of pins are listed below:

- **12V:** Connected to +ve of 9V battery.
- **GND:** Connected to GND.
- **5V:** Connected to a 5V source
- **ENA:** Motor enable pin. This is connected to a 5V source.
- **IN1:** Connected to pin A1(39) of ATmega32.

- **IN2:** Connected to pin A2(38) of ATmega32.
- **OUT1:** Connected to one end of DC motor.
- **OUT2:** Connected to the other end of DC motor.

We used a cardboard cutout with thin rope to simulate our floodgate with the DC Motor.

7.5 SPDT Switch

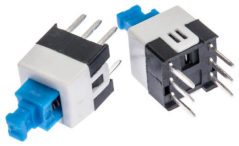

Figure 11: SPDT Switch

We used three SPDT switches. Switch MANUAL_AUTO was used to select between manual and auto mode. Switch GATE_UP and GATE_DOWN were used to pull the floodgate up and down respectively.
Connection of pins are listed below:

- **Throw which gets short when button is pushed down:** Connected to VCC for all three switches.
- **Throw which gets short when button is pushed up:** Connected to GND for all three switches.
- **Pole of MANUAL_AUTO switch:** Connected to pin A3(37) of ATmega32.
- **Pole of GATE_UP switch:** Connected to pin D3(17) of ATmega32.
- **Pole of GATE_DOWN switch:** Connected to pin B2(3) of ATmega32.

8 Challenges Faced

- **Shorting all GNDs:** First we used GND of different sources in different pin without connecting them together. Our system was not working as expected. Then we found that there was voltage difference between GND's of different sources. So, we shorted all GNDs and that solved our problem.

- **Interrupt:** We used external interrupt 1 and 2 to detect rising edge when Gate Up and Gate Down button are pressed respectively. But for a mysterious reason sometimes when Gate Up button was pressed ISR of interrupt 2 was being called though INT2 pin received no rising edge. Our ATmega32 was faulty. So when ISR for interrupt 2 was called we checked if INT2 pin had high input. This solved our problem. This was a temporary hack, a perfectly working ATmega32 chip should be used in this case.
- **CPU Frequency:** First we defined CPU frequency of ATmega32 at 16MHz as was defined in LCD codes we did earlier in our experiments. In that case there was significant difference in ATmega32's delay and real life delay. Later we set it at 1MHz and this problem was solved.
- **Faulty Wire:** During our partial project submission the sonar sensor stopped working. After trial and error for a long time, we detected a faulty wire. After this we borrowed a multimeter from our lab and used it until the final submission of our project. So, a multimeter should be used to solve this kind of problems efficiently.
- **Header Files:** To keep our large code manageable we created header files for sonar and motor operation which can be used in other projects in future.

9 Useful Links

Links Helpful for Interfacing:

- https://circuitdigest.com/microcontroller-projects/distance-measurement-using-hc-sr04-avr
- https://electrosome.com/interfacing-lcd-atmega32-microcontroller-atmel-studio/
- https://electrosome.com/interfacing-dc-motor-atmega32-l293d/

Project Material Links:

- **GitHub Repository:** https://github.com/tzpranto/Flood-Gate-Controller
- **Video Demonstration:**

10 Conclusion

This project helped us a lot to get acquainted with embedded systems and hardwares. The idea of this small scale project can be used on a larger scale to automatically control river dams. By proper modification and calculation, this idea can be used to connect all canals in Dhaka city and control their water flow

for better extraction of rain water. You are encouraged to use the materials attached in this report to expand this idea.