MEASURING VARIOUS PROPERTIES OF WATER IN RIVER USING DIFFERENT SENSORS AND A MICROPROCESSOR

A project report submitted in partial fulfillment of the requirements for the award

of the degree of

Bachelor of Technology in Electrical Engineering

By

Ch Omsrinath

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

TABLE OF CONTENTS

ABSTRACT

Rivers are the roots of all life. Civilizations flourished on the banks of rivers. They provide with necessary sustenance required for life. Due to recent developments, they are getting polluted more and more, being unable to support life or provide sustenance. There is a need to clean up the rivers for better survival of everyone. The fundamental part of cleaning up is assessing the damage. There is a need to gauge the various properties of water in the river throughout its length. With this data, it is possible to estimate parts of river which require higher care.

In this a new device capable of detecting various properties of water throughout the river is discussed. It is a hollow device containing a pH sensor, Electrical conductivity sensor, Temperature sensor and a GPS module connected to an Arduino. It also floats so that it can travel the length of river while measuring the required properties and saves them in a SD card so that it can be retrieved at the mouth of the river. These electronics are powered automatically with a wind generator located on the top of the device. It uses the wind speed over the surface of the river to generate electricity which is used to run the sensors. By using this device, we can get an accurate image of the water in the river.

CHAPTER 1

INTRODUCTION

It is very important to get accurate measurements of water characteristics in the river. We find these measurements using sensors connected to an arduino powered by wind speed over the surface of the river.

1.1 Motivation

Rivers are very long water bodies which exhibit different properties as the location varies. It is beneficial to measure those properties to fully capture the characteristics of the river. It becomes very cumbersome if we have to travel along the length of river to take readings at various points of the river. Hence, there is a need to measure these properties automatically to reduce man effort.

1.2 Overview and Problem Statement

The goal is to achieve a way of measuring the properties of the river automatically and get the data. There are several things to be taken into consideration. The biggest problem of making a device that is capable of measuring properties without sinking is power. Finding a way to powering such a device is a humongous task. It usually takes a very long time for the device to traverse the whole length of river. We can’t use batteries since they run out very quickly. Hence we need a renewable power source, that is capable of charging mid journey. Another major problems is handling the data. The data can’t be transmitted live because it requires tremendous energy. Hence we would have to store the data in te device itself and retrieve the data at the end of its journey. Since most electronics are not waterproof, we need a case to host the electronics. The case can’t be too heavy so that it would sink. It also can’t be too light so that the sensors don’t touch the water. It had to be designed in a way that it floats the right amount so sensors would be submerged. All these issues are to be dealt with.

CHAPTER 2

LITERATURE SURVEY

Water is very important for everyone. Water purity plays a vital role in many areas. Hence water quality testing is one of the major fields of measurements. There are several industrial grade water quality testers. These measure numerous properties of water such as, Dissolved oxygen, Temperature, pH, Turbidity, Conductivity, Salinity, Dissolved organic matter, Total suspended solids and etc. These testers are very accurate and can be used to to measure quality of water.

The major drawback of these testers are that they are high power equipments. Most of these are at higher end of power spectrum. They are also very heavy and each equipment have their own reader for measurement. They are not automatic and people had to carry them to place after place to take readings and manually save the data.

There are some devices available which take all the readings required automatically. These take the readings of the water and send them to us. Libelium smart kit water is one such device.

This device is capable of taking all the readings required and upload the data using cellular or long range zigbee protocol to the cloud. This is a low power device usually used in chemical detection and pollution level. It could sustain years with their battery. The only major drawback is that they are very costly and are typically used to monitor the properties at a certain location.

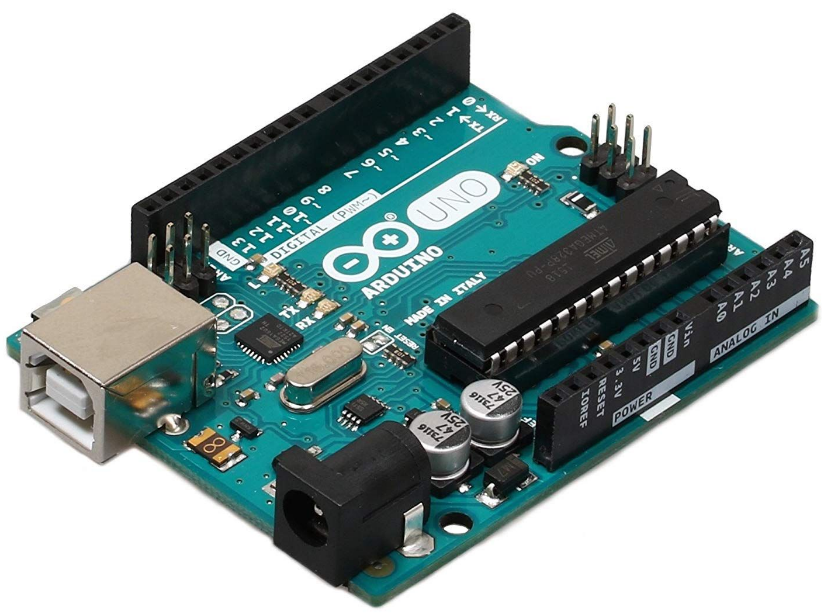
CHAPTER 3

PROPOSED WORK

3.1 ELECTRONICS

We use four different sensors namely, pH sensor, Electrical Conductivity sensor, Temperature sensor and a gps module. These sensors are connected to a microprocessor, Arduino to process the information and store them in a SD card, using a sd card module, so that the readings could be retreated later.

Arduino: Arduino is an open source platform that facilitates easy building of electronic projects. An Arduino board consists of a microprocessor chip embedded in it.



Arduino uno, the variant used in this project has ATMEGA328P as its microprocessor. It is capable of supporting 14 digital I/O and 6 Analog input pins. It also has a USB port to program the board, a power jack to power it and a reset button. Arduino Uno can be programmed with Arduino software. ATMEGA 328P is equipped with a bootloader so that the code can be uploaded directly.

pH Sensor: pH is a measure of acidity or alkalinity of a solution, the pH scale ranges from 0 to 14. The pH indicates the concentration of hydrogen ions present in certain solutions. It can accurately be quantified by a sensor that measures the potential difference between two electrodes: a reference electrode (silver / silver chloride) and a glass electrode that is sensitive to hydrogen ion.

This signal can be captured accurately using Arduino and is properly scaled up to give an accurate reading of pH. It is connected to an analog pin in arduino, through which the reading will be taken.

Electrical Conductivity sensor: Conductivity is an important parameter of water quality. It can reflect the extent of electrolytes present in water. Electrical Conductivity sensor is specially used to measure the electrical conductivity of aqueous solution, and then to evaluate the water quality. Conductivity is the reciprocal of the resistance, which is related to the ability of the material to carry the current. In the liquid, the reciprocal of the resistance, the conductivity, is the measure of its ability to conduct electricity.

It supports 3~5v wide voltage input, and is compatible with 5V and 3.3V main control board; The output signal filtered by hardware has low jitters; The excitation source adopts AC signal, which effectively reduces the polarization effect, improves the precision and prolongs the life of the probe; The software library uses two-point calibration method, and can automatically identify standard buffer solution. It is also connected to an analog pin of the Arduino. For better readings of this sensor, it is used together with temperature sensor.

Temperature sensor: The **DS18B20** is a 1-wire programmable Temperature sensor from maxim integrated. It is widely used to measure temperature in hard environments like in chemical solutions, mines or soil etc. The constriction of the sensor is rugged and also can be purchased with a waterproof option making the mounting process easy.

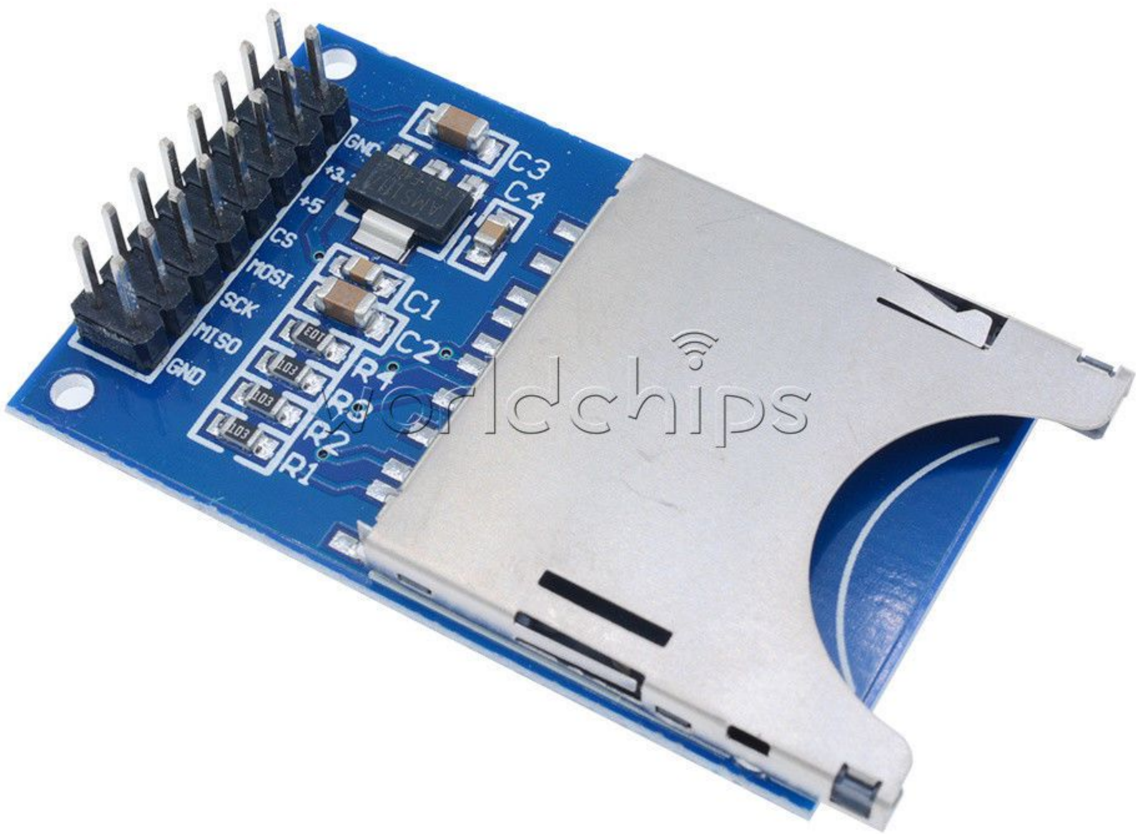
It can measure a wide range of temperature from **-55°C to +125°** with a decent accuracy of **±5°C**. Each sensor has a unique address and requires only one pin of the MCU to transfer data so it a very good choice for measuring temperature at multiple points without compromising much of your digital pins on the microcontroller. This sensor has three pins. Vin, Gnd and data pin.

GPS Module: At the heart of the module is a NEO-6M GPS chip.It can track up to 22 satellites on 50 channels and achieves the industry's highest level of sensitivity i.e. -161 dB tracking, while consuming only 45mA supply current. It can do up to 5 location updates a second with 2.5m Horizontal position accuracy. It also boasts a **Time-To-First-Fix** (TTFF) of under 1 second. It allows a reduction in system power consumption by selectively switching parts of the receiver ON and OFF. This dramatically reduces power consumption of the module to just **11mA** making it suitable for power sensitive applications like GPS wristwatch.The necessary data pins of NEO-6M GPS chip are broken out to a 0.1″ pitch headers. This includes pins required for communication with a microcontroller over UART. The module supports baud rate from 4800bps to 23400bps with default baud of 9600.

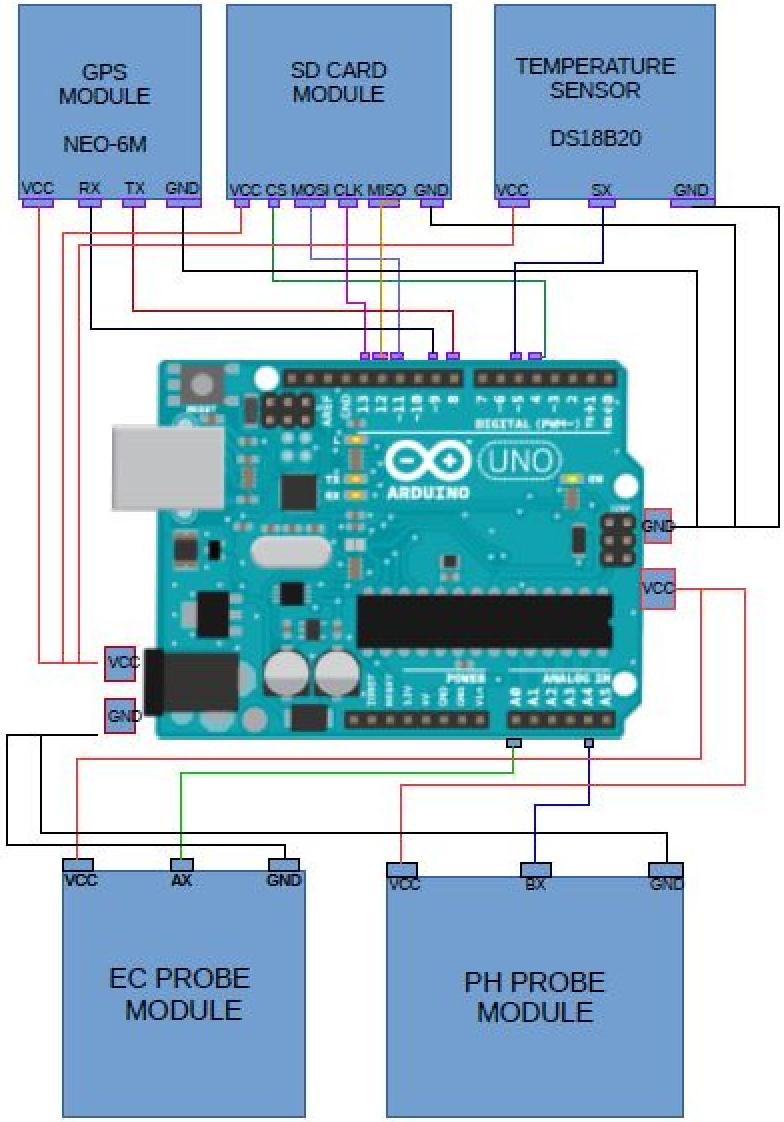
The module is equipped with an **HK24C32** two wire serial EEPROM. It is 4KB in size and connected to the NEO-6M chip via I2C. It also contains a rechargeable button battery which acts as a super-capacitor. An EEPROM together with battery helps retain the **battery backed RAM** (BBR). The BBR contains clock data, latest position data(GNSS orbit data) and module configuration. But it’s not meant for permanent data storage. As the battery retains clock and last position, time to first fix (TTFF) significantly reduces to 1s. This allows much faster position locks.

Without the battery the GPS always cold-start so the initial GPS lock takes more time. The battery is automatically charged when power is applied and maintains data for upto two weeks without power. It has four pins of which two are power pins. The remaining two are Transmission and Receiving pins.

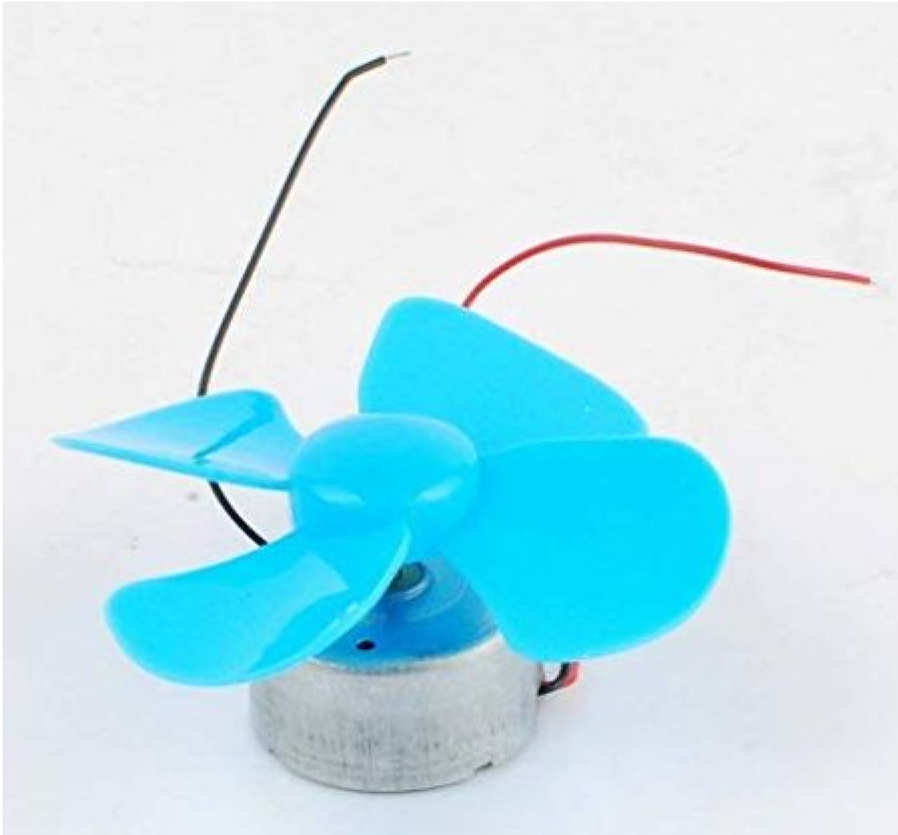
SD Card module: This module uses the standard SPI interface for communication, which involve SPI buses, MISO, MOSI, SCK, and a CS signal pin. through programming, the data can easily be read and written into SD Card by using the Arduino or other microcontrollers.



Together, they measure various properties of water in the river. The final circuit connecting all these different sensors is as



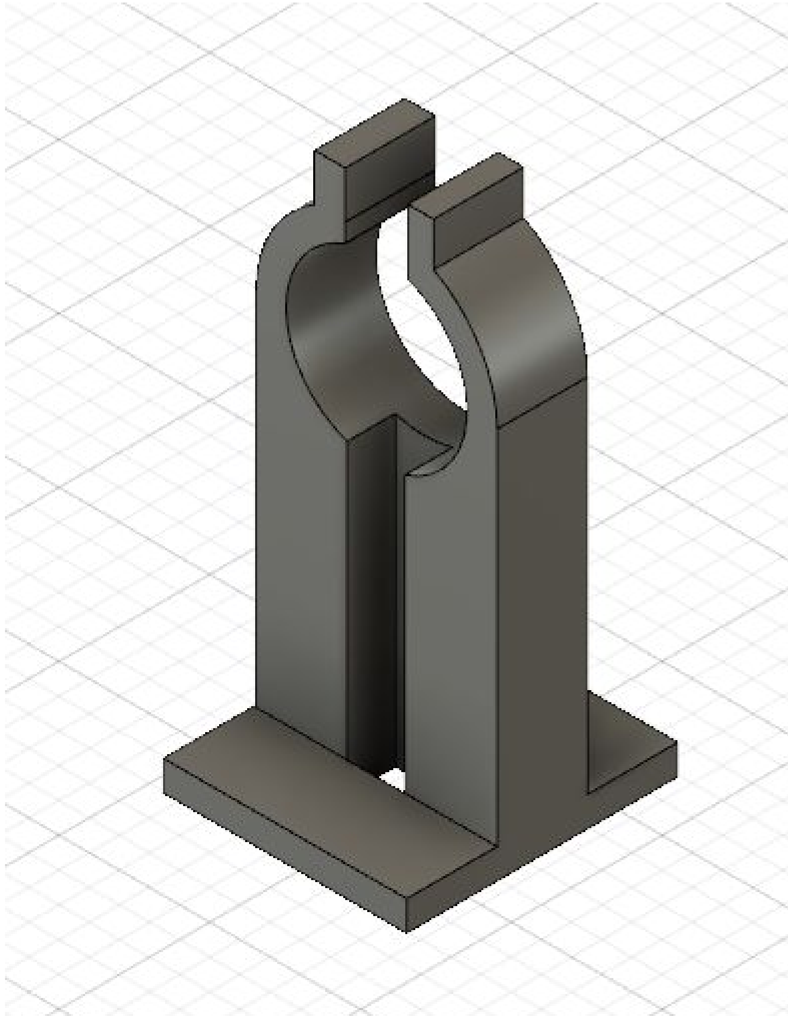
Electrical generator: Instead of using electricity to turn a rotor, generator generates electricity by turning the rotor. Winds over the river, turn a turbine and is converted to actual power through which the sensors and the arduino is powered.



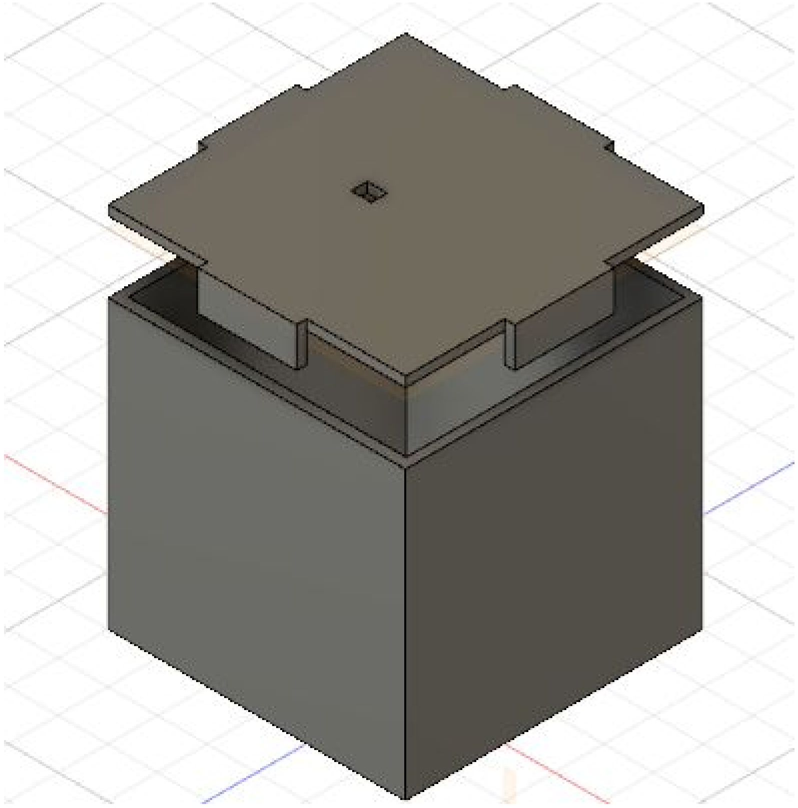
3.2 MECHANICS

All the sensors and the Arduino are housed within a hollow cube, which has a top designed to case a wind generator and a bottom designed to make it float.

Top: It is located above the hollow cube and has the structure to support the wind generator. It has a 4cm x 4cm approximate size of the base and can be clamped by the supports on the side.



Cube: It is a hollow cube with a top lid. It has holes on the sides,through which sensors get in the contact with the water. It's hollow ness makes the cube float.



Bottom: It is a hollow hemisphere which can be filled with varying amounts of weights. The weight of filling depends on the level of submerge of the whole box. This way it floats on the water and travels across the river.

CHAPTER 4

RESULT

Power Calculation:

All the power calculations are done at 5V.

Sensor	Avg Power	Max Power
SD Card module	80mA	200mA
Temperature Sensor	5mA	
GPS Module	45mA	
EC and pH probe	30mA	
Arduino	25mA	36mA

Overall average power consumption of the circuit is observed at 185mA.

The power consumption keeps varying and the maximum consumption is observed at 316mA.

For, 5V operating voltage,

Average Power = 925mW and

Maximum Power = 1580mW

Now, using a wind generator, we have a motor with a rated wind speed of 3m/s with a rated voltage of 12V with max power generation of 200mA. Avg wind speed over the river is clocked at 2m/s. The equivalent power that is generated with an average speed of 2m/s is,

P = 12x200x3/2mW

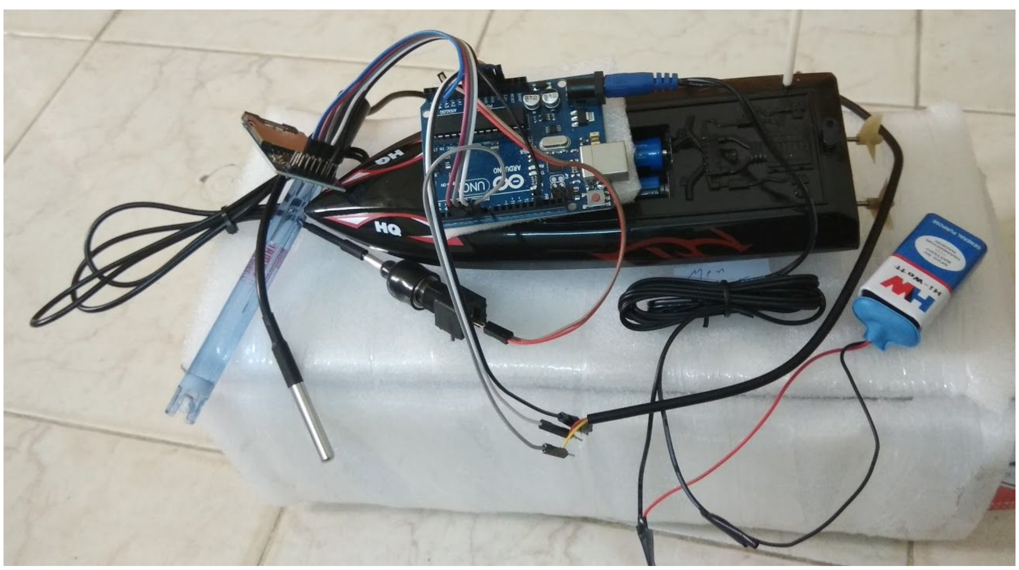
Total power = 3600mW

It can be observed that total power generated is much more than required. The DC generator can sustain the whole circuit.

Feasibility

To test the feasibility of the proposed work, a small model to validate the electronics has been prepared. Two sensors, namely, pH sensor and temperature sensor are attached to arduino and are tested in water. The data is saved in SD Card.

The circuit is attached to water boat, so that it can be tested against winds and various water samples. However, pH sensor that is chosen is very sensitive and it takes around a minute for a stable reading. This might result in error if it is used to take a reading while in motion in small water bodies.



Temperature sensor:

The code for Temperature Sensor is available as on open source project for Arduino

```
/***************************************************************
*****/
// First we include the libraries
#include <OneWire.h>
#include <DallasTemperature.h>
/***************************************************************
*****/
// Data wire is plugged into pin 2 on the Arduino
#define ONE_WIRE_BUS 2
/***************************************************************
*****/
// Setup a oneWire instance to communicate with any OneWire
devices
// (not just Maxim/Dallas temperature ICs)
OneWire oneWire(ONE_WIRE_BUS);
/***************************************************************
*****/
// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);
/***************************************************************
*****/
```

```
void setup(void)

{

 // start serial port

 Serial.begin(9600);

 Serial.println("Dallas Temperature IC Control Library Demo");

 // Start up the library

 sensors.begin();

}

void loop(void)

{

 // call sensors.requestTemperatures() to issue a global

temperature

 // request to all devices on the bus

/****************************************************************

*****/

 Serial.print(" Requesting temperatures...");

 sensors.requestTemperatures(); // Send the command to get

temperature readings

 Serial.println("DONE");

/****************************************************************

*****/

 Serial.print("Temperature is: ");
```

```
 Serial.print(sensors.getTempCByIndex(0)); // Why "byIndex"?

   // You can have more than one DS18B20 on the same bus.

   // 0 refers to the first IC on the wire

   delay(1000);

}
```

The result is as,

```
Dallas Temperature IC Control Library Demo
 Requesting temperatures...DONE
Temperature is: 34.00 Requesting temperatures...DONE
Temperature is: 34.00 Requesting temperatures...DONE
Temperature is: 34.00 Requesting temperatures...DONE
Temperature is: 34.00 Requesting temperatures...DONE
Temperature is: 34.06 Requesting temperatures...DONE
Temperature is: 34.19 Requesting temperatures...DONE
Temperature is: 34.31 Requesting temperatures...DONE
Temperature is: 34.44 Requesting temperatures...DONE
Temperature is: 34.50 Requesting temperatures...DONE
Temperature is: 34.56 Requesting temperatures...DONE
Temperature is: 34.50 Requesting temperatures...DONE
Temperature is: 34.50 Requesting temperatures...DONE
Temperature is: 34.50 Requesting temperatures...DONE
Temperature is: 34.50 Requesting temperatures...DONE
Temperature is: 34.50 Requesting temperatures...DONE
Temperature is: 34.50 Requesting temperatures...DONE
Temperature is: 34.44 Requesting temperatures...DONE
Temperature is: 34.44 Requesting temperatures...DONE
Temperature is: 34.50 Requesting temperatures...DONE
Temperature is: 34.56 Requesting temperatures...DONE
Temperature is: 34.56 Requesting temperatures...DONE
Temperature is: 34.56 Requesting temperatures...DONE
```

For the dc generator,

This example code is in the public domain.

```
 http://www.arduino.cc/en/Tutorial/AnalogInput

 */
int sensorPin1 = A1;
//int sensorPin2 = A2;    // select the input pin for the
potentiometer
int ledPin = 13;      // select the pin for the LED
int sensorValue1 = 0; // variable to store the value coming from
the sensor
int sensorValue2 = 0;
void setup() {
  // declare the ledPin as an OUTPUT:
  //pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // read the value from the sensor:
  sensorValue1 = analogRead(sensorPin1);
  //sensorValue2 = analogRead(sensorPin2);
  Serial.println(sensorValue1*12.0/1024.0);
  // turn the ledPin on
```

```
 // digitalWrite(ledPin, HIGH);

  // stop the program for <sensorValue> milliseconds:

 // delay(sensorValue);

  // turn the ledPin off:

  //digitalWrite(ledPin, LOW);

  // stop the program for for <sensorValue> milliseconds:

  delay(1000);

}
```

The result for a normal slow wind is around 4 V

For an sd card module sample code is already available

```
////////////////////////////////////////////////////////////////
/
   //              Arduino SD Card Tutorial     v1.00
//
  //       Get the latest version of the code here:
//
 //         http://educ8s.tv/arduino-sd-card-tutorial
//
////////////////////////////////////////////////////////////////
/
#include <SD.h>
#include <SPI.h>
int CS_PIN = 10;
File file;
void setup()
{
  Serial.begin(9600);
  initializeSD();
  createFile("test.txt");
  writeToFile("This is voltage and temperature!");
  closeFile();
```

```
  openFile("prefs.txt");
  Serial.println(readLine());
  Serial.println(readLine());
  closeFile();
}
void loop()
{
}
void initializeSD()
{
  Serial.println("Initializing SD card...");
  pinMode(CS_PIN, OUTPUT);
  if (SD.begin())
  {
    Serial.println("SD card is ready to use.");
  } else
  {
    Serial.println("SD card initialization failed");
    return;
  }
}
int createFile(char filename[])
{
  file = SD.open(filename, FILE_WRITE);
  if (file)
  {
    Serial.println("File created successfully.");
    return 1;
  } else
  {
    Serial.println("Error while creating file.");
    return 0;
  }
}
int writeToFile(char text[])
{
  if (file)
  {
    file.println(text);
```

```
    Serial.println("Writing to file: ");
    Serial.println(Value);
    return 1;
  } else
  {
    Serial.println("Couldn't write to file");
    return 0;
  }
}
void closeFile()
{
  if (file)
  {
    file.close();
    Serial.println("File closed");
  }
}
int openFile(char filename[])
{
  file = SD.open(filename);
  if (file)
  {
    Serial.println("File opened with success!");
    return 1;
  } else
  {
    Serial.println("Error opening file...");
    return 0;
  }
}
String readLine()
{
  String received = "";
  char ch;
  while (file.available())
  {
    ch = file.read();
    if (ch == '\n')
    {
```

```
      return String(received);
    }
    else
    {
      received += ch;
    }
  }
  return "";
}
```

This way, the sensors can be used to calculate properties of river.

CHAPTER 5

CONCLUSION

The device is capable of measuring different properties of water. The power generated from dc generator is enough to sustain the whole set. However, it is not advisable to connect the voltage from the generator directly to power Arduino. It needs a stable voltage for better performance. Alternative, would be to power only Arduino using a battery whilst the generator is supporting the rest of circuitry.

This has a lot of scope for improvement in the design aspect. The hollow cube design is basic. It can be further improved by making it more streamlined to the course of river. The absence of a stable voltage for Arduino is to be rectified. A dual battery system, in which one is recharged while the other is being used might help to achieve a more stable solution. If the power generation is stabilised, a transmission and receiver system can also be incorporated which reduces the need to store the data in the system.

CHAPTER 6

REFERENCES

https://store.arduino.cc/usa/arduino-uno-rev3

https://scidle.com/how-to-use-a-ph-sensor-with-arduino/

https://www.dfrobot.com/product-1123.html

https://components101.com/sensors/ds18b20-temperature-sensor

https://lastminuteengineers.com/neo6m-gps-arduino-tutorial/

https://www.instructables.com/id/Micro-SD-Card-Tutorial/

http://www.electronics-lab.com/project/interfacing-arduino-micro-sd-card-module/

https://create.arduino.cc/projecthub/TheGadgetBoy/ds18b20-digital-temperature-sensor-and-arduino-9cc806