*Journal of Electrical and Electronics Engineering*
Vol. 2, No. 2, December 2018, 54-59

# Automatic Wireless Mobile Weather Station

Vincent, Enndi Chiu, Kim Sung Tae, Ikhsan Bukhori, Surawan Setiyadi
Department of Electrical Engineering, President University, Bekasi 17550, Indonesia
**Email:** vincentsudiono@outlook.com, enndichiu@gmail.com, tjdxodia91@gmail.com, iksan.bukhori@president.ac.id, surawan@president.ac.id
**Corresponding author:** vincentsudiono@outlook.com

*ABSTRACT*

**In the electronic devices, there are so many functions that we can use to make life easier. In the city especially, nearly all of them use technology as the system, it can also be used to store data, measuring data to help the others quantitative and qualitatively. One of the things that is very useful is the weather station, where the weather station used in the city to maintain the humidity, the intensity of the light, and the temperature. But in the rural areas, somehow the weather station is not available because of the limitation of the technology itself. And also, sometimes the place that we want to measure with is quite dangerous because there are so many predators. Because of that, we've come up to the new brand technology called Automatic Wireless Mobile Weather Station. This device helps us to measure the rural areas, dangerous places, using robot with wireless connection with weather station features. This device can be implemented into a moving robot and can also send us the data to the mobile phones through Bluetooth technology, and also it can measure the humidity, temperature, and the intensity of the light (unit of lux). Finally, the data can be controlled 24 hours through wireless connection, so we don't really need to go into the location because we can control it using this Automatic Wireless Mobile Weather Station.**

**Keywords: Bluetooth, LDR, DHT11, Arduino, L298N, LM2956, Weather Station.**

## I. Introduction

Weather is essential because it contains some information such as the humidity, temperature, and intensity of the light. Those three parameters are correlating with each other and cannot be separated. Intensity of the light will affect the temperature, and the temperature will affect the humidity, so all correlates with each other.

When reading a data, a noise is one of the most serious problem. It may distort the signal that we would like to measure. A filter helps us to protect the signal still in shape. The most common one as easy to implement is the RC filter, since the size is relatively smallest beside another filter. The humidity and temperature sensors have their own internal filter itself, to filter the noise. While the LDR does not have its own filter.

Actually, to measure the it, people usually built a weather station that can't be moved because it is placed inside the building to measure. But in some rural areas, we need something that can be controlled to measure the humidity, temperature, and intensity of the light. And we also need to store the data using wireless connection because it is unlikely the static weather station.

This project is meant to read the weather by measuring the temperature, humidity, and the light intensity of a certain location. This reading will be done every second. But this measurement can be in the area that cannot be reached by humans, or maybe it's too dangerous for humans, so having a robot car with weather station attached to it will be helpful. The control is done from the other place, and also the data reading. This paper consists of 8 parts. Starting with the introduction, the second one is system design which cover some basic concepts, the third is the hardware implementation where all the design of the circuit are drawn. The fourth part is the software implementation, where the things that we have done is being analyzed using the point of view. And the fifth part will be the result and discuss some errors of this project. For the sixth part will be about conclusion, and the last part will be discussing the prospects for this kind of development.

## II. LITERATURE REVIEW

### *A. Arduino*

Arduino is an open hardware development board that can be used by makers to design and can build many kinds of the devices that interact with the real world. While Arduino refers to a specific type of board design, it can also be used to refer to a company which manufactures a specific implementation of these boards, also we can make the any measurement with the Arduino. In this case we are using it to control the Robot and installed the measurement of lights and temperature. An example of this board is shown in Figure 1.

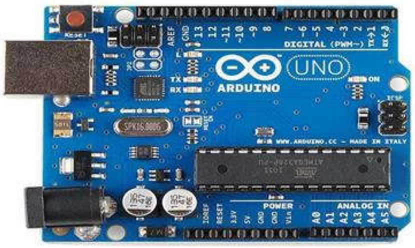


Figure 1. Arduino Uno R3.

Received: 9 February 2020 Revised: 9 March 2020 Accepted: 2 April 2020

### *B. Light Dependent Resistor (LDR)*

The LDR (Figure 2) is a special resistor which will have a non-fixed value of resistance. Basically, it is a semiconductor, which have a conductance between a conductor and an insulator.

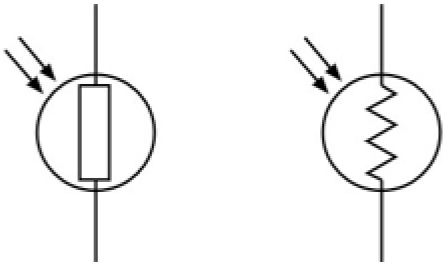


Figure 2. Light Dependent Resistor (LDR) symbol..

The value of its resistance is depending to the light intensity. Exposing it to a light will make the photon to give energy for the electrons from the valence band to be excited into the conductance band. The more intense light means more photon, and more photon means more energy. Thus, with more energy we will have more electrons get excited to the conduction band. More electrons mean that we have higher conductance, and in other words, we have lower resistance. To sum up, the brighter the light, the less resistance we have. And the other case is if the light intensity is relatively low, it will make a higher resistance.

During the experiment done in the Electrical Engineering Laboratory, data for the resistance of the LDR at three different light condition is shown in Table 1.

**Table 1. LDR resistance for Some Light Condition**

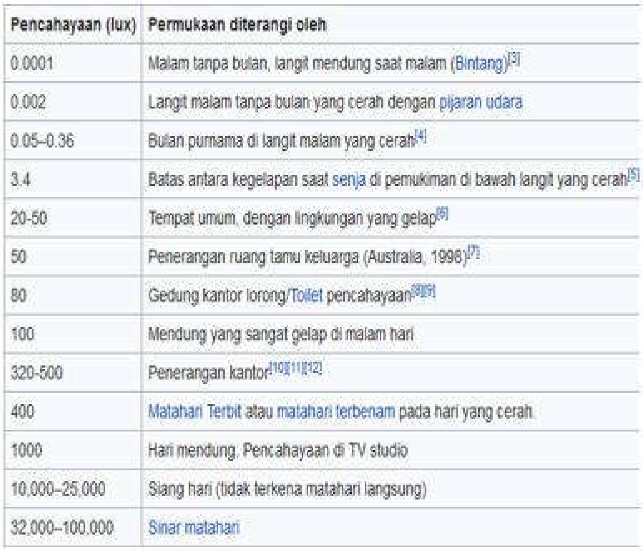
| Light Condition | Resistance (Ω) |
|---|---|
| Bright | 20 |
| Normal | 1.000 |
| Dark | 5.000 |

Where the bright condition is achieved by using the flashlight from the hand phone and place at a very close distance to the surface of the LDR. And the dark condition is achieved by totally closing the surface of the LDR by using hand.

### *C. Light Intensity*

The light (sometimes referred as photon) has some terms of description. The flux is a term used to describe the rate of energy flow. But a more specific term is a luminous flux, which is the measure of the power of visible light, considering a light to have some frequencies, and one of them is the visible light. Luminous flux has the unit called lumens. Then we also have the illuminance as a measure of flux per unit area. It can also be called as visible flux density. Illuminance can also be referred as visible light intensity, or light intensity in short, and it is expressed in lux (lumens per square meter).

Some values of illuminance can be seen in the Figure 3.



| Pencahayaan (lux) | Permukaan diterangi oleh |
|---|---|
| 0.0001 | Malam tanpa bulan, langit mendung saat malam (Bintang)[3] |
| 0.002 | Langit malam tanpa bulan yang cerah dengan pijaran udara |
| 0.05–0.36 | Bulan purnama di langit malam yang cerah[4] |
| 3.4 | Batas antara kegelapan saat senja di pemukiman di bawah langit yang cerah[5] |
| 20-50 | Tempat umum, dengan lingkungan yang gelap[6] |
| 50 | Penerangan ruang tamu keluarga (Australia, 1998)[7] |
| 80 | Gedung kantor lorong/Toilet pencahayaan[8][9] |
| 100 | Mendung yang sangat gelap di malam hari |
| 320-500 | Penerangan kantor[10][11][12] |
| 400 | Matahari Terbit atau matahari terbenam pada hari yang cerah |
| 1000 | Hari mendung, Pencahayaan di TV studio |
| 10,000–25,000 | Siang hari (tidak terkena matahari langsung) |
| 32,000–100,000 | Sinar matahari |

Figure 3. Some values of illuminance

### *D. HC-05*

HC-05 module is an easy to use Bluetooth module, designed for transparent wireless serial connection setup. Serial port Bluetooth module is fully qualified Bluetooth V2.0 +EDR (Enhanced Data Rate) 3 Mbps modulation with complete 2.4Ghz radial transceiver and baseband. It uses CSR Blue core 04.

External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency hopping Feature). And it has two operating modes, one is the data mode in which it can send and receive data from other Bluetooth devices and the other is the AT command mode where the default device setting can be changed. We can operate the device in either of these two modes by using the key pin (see Figure 4).

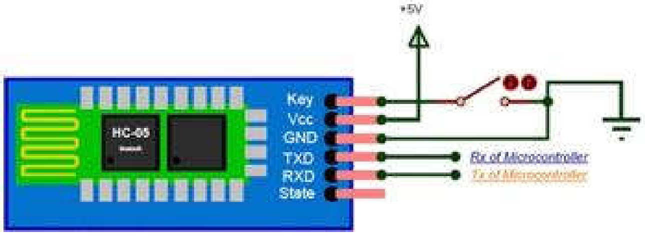


Figure 4. HC-05 Bluetooth module.

### *E. Piezo Speaker*

A piezo speaker (see figure 5), also known as buzzer, is basically a tiny speaker that we can connect directly to and Arduino. From the Arduino, we can make sounds with a buzzer by using tone. It can be found in alarm devices, computers, timers, and confirmation of user input such as a keystroke.

Figure 5. Piezo Speaker(buzzer).

### F. *DHT-11*

With the DHT11 module we can do the measurement of temperature and relative humidity with the Arduino. These sensors contain a chip that does analog to digital conversion and spit out a digital signal with the temperature and humidity. With this component we can make easily to use with microcontroller. The physical appearance of this sensor can be seen in Figure 6.

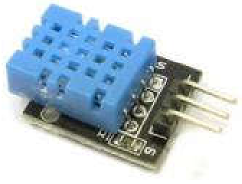


Figure 6. DHT-11 Sensor.

It can hold the temperature range 0 to 50 C, Humidity range 20 to 90%, and the operating voltage 3 – 5.5 V DC with 0.5-2.5 mA

### G. *LM2596*

The LM2596 is a DC-DC converter. This component will do the step down from to DC voltage, and giving its name as the step-down DC to DC converter. This component is useful especially when we need a constant regulated DC power supply for the electrical components.

The concept of doing it is by turning on and off a switch in a circuit of a serial-connected voltage source, inductor, diode, and a capacitor. By turning on the circuit, the inductor and the capacitor will be charged, and when it is turned off, the capacitor will act like the new voltage source with a regulated voltage. The period of turning the switch on and off will determine the step-down conversion of the voltage. Besides having a regulated voltage, the output will have a constant voltage at the same time, thus, giving us a constant regulated DC power supply for the electrical components. The connection for LM2596 is shown in Figure 7.

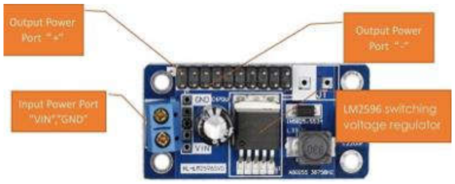

Output Power Port "+"
Output Power Port "-"
Input Power Port "VIN","GND"
LM2596 switching voltage regulator

Figure 7. Connection for LM2596.

### H. *Sensor Shield*

The LM2596 is a DC-DC converter. This component will do the step down from to DC voltage, and giving its name as the step-down DC to DC converter. This component is useful especially when we need a constant regulated DC power supply for the electrical components.

One of the most useful components for Arduino is a shield. It provides the user an easier way to connect the desired parts for the project. There are some kinds of shield, and one of them is the sensor shield. It provides 13 digital pins and 6 analogue pins, each having its own voltage and the ground port. It also has special ports for the Liquid Crystal Display (LCD), ultrasonic pin, and some other components. By having this shield, we can connect each component we use without having to combine the ground pins or the voltage pins. We can easily connect each component without disturbing the other, and we could also add more components without having the other detached. This shield is shown in Figure 8.

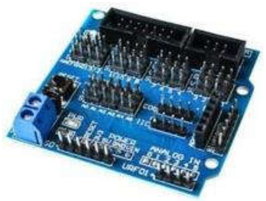


Figure 8. Sensor Shield for Arduino Uno.

### I. *L298N*

The DC motor speed and polarity can be controlled using the DC motor controller, and one of them is L298N (see Figure 9). This motor driver has two ports for motor, which means it can control maximum 2 DC motors. The way to control the speed is by using the Pulse Width Modulation for the voltage, in which will convert a certain value of voltage to an exact-valued lower voltage. A lower voltage inputted to the DC motor means a slower speed, while a higher one means faster. This controlling is done via pin EnA and EnB in the port of L298N. Giving an analogue signal with specific value will be enough to control the speed. Besides, controlling the movement of the DC motor, either it is forward or backward, can be done by having the polarity changed.

The connection to each DC motor has two ports. Having one as positive while the other one is negative will make it turn forward or backward, while giving the opposite polarity will make it the other way. For motor on the left is controlled by En1 and En2, while motor on the right is controlled by En3 and En4. In practical, this can be done simply by assigning values of digital signal as 1 (for positive), or 0 (for negative).

Figure 9. L298N DC motor driver..

### J. *Switch Module*

The switch module is to provide a way to connect and disconnect two connections. Basically, it acts like a normal switch, but having a better appearance and ports for connections. The button is a toggle-type, where pressing it once will have the button to be in the on position. And pressing it again will make the button to be in the off position. This is very helpful for a project having its power source from the battery, where we can turn off the system by pressing the button and let no connection unplugged.

## III. Implementation

### A. Hardware Implementation

The project will be implemented in the circuit as seen in Figure 10. The temperature and humidity measurement will be done by the DHT11 module. The light intensity measurement is done by the LDR, in which is connected in a circuit of voltage divider. Reading the voltage of the resistor accompanying the LDR can be used for this, since the changing of light intensity means changing resistance of LDR, and it will also change the value read. Both pins for these reading will go to the analogue pins, since we must have a specific value. Besides, the robot car will have two DC motors at the hind wheel, and a servo controlling the front wheel for turning. The DC motor is connected to the L298N as the driver. This driver is then connected to the Arduino. Be careful for the connection of the inputs of this motor driver. Make sure that the input is coming from the switch module instead from the DC to DC converter, because we want to make sure that the driver will only start when the switch is on. Then the LED will then be connected to the Arduino, and also the buzzer. Technically, all of the components will be connected to the sensor shield.

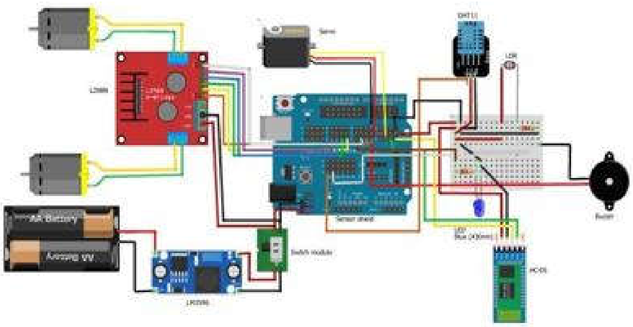


Figure 10. Circuit diagram.

The pinout can be seen in Table 2.

**Table 2. Pins Connection**

| Component | Pin |
|---|---|
| DHT11 | A0 |
| LDR | A1 |
| LED | 7 |
| Buzzer | 4 |
| HC-05 | Rx – 1<br>Tx – 0 |
| Servo | 2 |
| L298N | EnA – 5<br>EnB – 6<br>En1 – 8<br>En2 – 9<br>En3 – 10<br>En4 – 11 |

### B. Software Implementation

The circuit which has been designed is then programmed such that it will work as what we want it to be. The combined circuits will have to be able to take the data of temperature, humidity, and light intensity every second and send it via Bluetooth, while at the same time read the input from the user for the movement of the robot car. The movement must not disturb the data reading for the weather measurement. So, we are attaching the interrupt for the Bluetooth reading. We are using the pin 2 as the interrupt, but actually, we have used pin 2 for the servo. The interrupt is also made to be in mode of change, it will be triggered if there are any changes in the signal value. So, it means that any changes to the value of servo will trigger the Interrupt Service Routine (ISR). While in real application, the servo is trying to maintain its position, and making it having the value changes all the time. Thus, we will have the ISR to be accessed at any time, and ready for reading. This way, we can let the HC-05 to focus on the communication, instead of also having it to be giving a signal for the interrupt. Besides having all of these features, we also have the buzzer to be turned on as an indicator of a hot environment, and an LED to be turned on as an indicator of a dark environment. The threshold values for each of them is stated in the program, where any value above 39 degrees will trigger the buzzer, and the value below 51 lux will turn on the LED.

The complete Arduino code used in this project is shown in Table 3.

**Table 3. Arduino Code**

```
#include <SoftwareSerial.h>
#include <dht.h>
#include<Servo.h>
Servo servo;
SoftwareSerial module_bluetooth(0, 1); // pin RX | TX int
EnA=5, EnB=6;
int En1=8, En2=9, En3=10, En4=11;
int forward = 1; int backward = -1; int turnR = 2;
int turnL = -2;
int s = 500; //fastet speed
char command = '0'; //taking the command from
bluetooth dht DHT;
int dht_pin = A0;
int temp_threshold = 40; int ldr_pin = A1;
double ldr_val = 0; double lux = 0;
int lux_threshold = 50; int relay_pin = 7;
bool led_status = false; int buz_pin = 4;
int buz_tone = 5000;
int buz_delay = 500;
bool buz_status = false;

void setup() {
Serial.begin(9600);

module_bluetooth.begin(9600);
servo.attach(2);
servo.write(90); //default position
pinMode(EnA, OUTPUT); //left, speed control
pinMode(En1, OUTPUT); //left, forward
pinMode(En2, OUTPUT); //left, backward
pinMode(EnB, OUTPUT); //right, speed control
pinMode(En3, OUTPUT); //right, forward
pinMode(En4, OUTPUT); //right, backward
pinMode(dht_pin, INPUT);
```

```
   pinMode(ldr_pin, INPUT);
   pinMode(buz_pin, OUTPUT);
   pinMode(relay_pin, OUTPUT);
    Serial.println("\nTemp Hmd Lux Buz LED");
    attachInterrupt(0, gogo, CHANGE);
  }
  void loop() {
    ldr_val = analogRead(ldr_pin);
    lux = (788.69)*(pow((ldr_val/(1023-ldr_val)),(1.4)));
    //converting the signal into lux
    DHT.read11(dht_pin); //reading the data of temperature and
                                    humidity
    if(DHT.temperature >= temp_threshold){
    tone(buz_pin, buz_tone, buz_delay);
    buz_status = true;
    }else{
      noTone(buz_pin);
      buz_status =
      false;
    }
    if(lux <= lux_threshold){
      digitalWrite(relay_pin,
      HIGH); led_status = true;
    }else{
      digitalWrite(relay_pin,
      LOW); led_status = false;
    }
    //printing out all of the values
    Serial.print(DHT.temperature);
    Serial.print(" ");
    Serial.print(DHT.humidity);
    Serial.print(" "); Serial.print((int) lux);
    Serial.print(" ");

    if(buz_status){
      Serial.print("ON");
    }else{
      Serial.print("OFF"
      );
    }
    Serial.print(" ");
    if(led_status){
    Serial.println("ON");
    }else{
      Serial.println("OF
      F");
    }
    delay(1000); //1 second delay before the next reading
  }
  void gogo(){ //ISR
    if(Serial.available()>0){ command =
    Serial.read(); Serial.print("
    command : ");
    Serial.println(command);
    switch(command){
        case 'F' : go(forward, s);
        break; case 'B' : go(backward,
        s); break; case 'R' : go(turnR,
        s); break;
        case 'L' : go(turnL, s); break;
        case 'S' : go(forward, 0); break; //stop default :
        go(forward, 0); break; //stop
      }
      delay(10);
    }
  }
  void tilt(int value){
    servo.write(value);
  }
  void go(int dir, int speed1){
    switch(dir){
      case   1   :   digitalWrite(En1,   1);   digitalWrite(En2,   0);
```

```
digitalWrite(En3, 1);
               digitalWrite(En4, 0); tilt(90); break; //forward


      case -1 : digitalWrite(En1, 0); digitalWrite(En2, 1);
                  digitalWrite(En3,
      0); digitalWrite(En4, 1); tilt(90); break; //backward

      case 2 : digitalWrite(En1, 1); digitalWrite(En2, 0);
               digitalWrite(En3, 1); digitalWrite(En4, 0); tilt(70); break;
               //turn right, then set the servo bcak to normal position
      case -2 : digitalWrite(En1, 1); digitalWrite(En2, 0);
               digitalWrite(En3, 1); digitalWrite(En4, 0); tilt(120); break;
               //turn left, then set the servo bcak to normal position
      default : digitalWrite(En1, 0); digitalWrite(En2, 0); digitalWrite(En3,
      0);
               digitalWrite(En4, 0); break;
    }
    analogWrite(EnA, speed1);
    analogWrite(EnB, speed1);
}
```

## IV. Result

### *A. Result*

The result of this project is connected to a chassis for the robot car as shown in Figure 11.

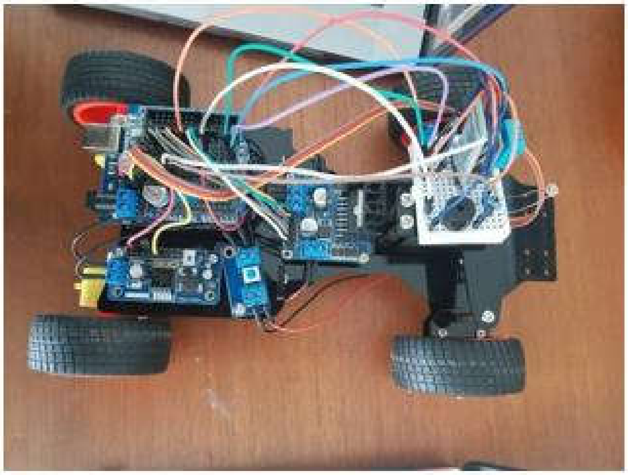


Figure 11. Hardware Installation.

The voltage source is coming from two batteries with 3.7 V each, and thus providing 7.4 V which goes regulated by the DC to DC converter before being used by the Arduino and other components. The controlling is done by using the application from the smartphone. The interface can be seen in Figure 12.

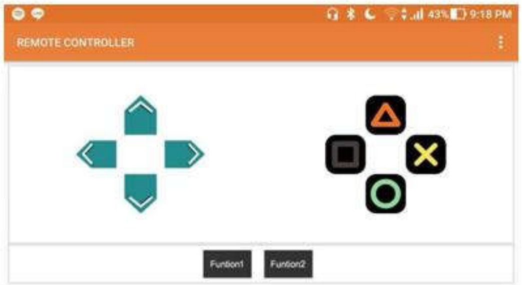


Figure 12. Controller Interface.

The button we use is just the up, down, left, right, and the rectangular button. The up button will give a character of ‘F’, the right will give ‘R’, left will give ‘L’, and down will give ‘B’ to the Arduino. Each character is then representing one movement for the robot car. While the rectangular button gives ‘S’, which is meant for emergency stop, in case of the robot does not stop.

Besides, the data taken by the weather station can be seen from the same application with different interface. The output looks like the serial monitor in the Arduino IDE, but with some different view, since it is in the smartphone. The interface can be seen in Figure 13.

Figure 13. Data display Interface.

### B. *Analysis*

In the real-life application, the Bluetooth module will have the connection with the user’s smartphone in a certain range only. As the distance get further, there will be a certain noticeable delay due to the distance that has to be taken by the signal is longer than before. But as we have experimented, the connection is still good with no noticeable delay for a distance below 10 m. But further than 10 m, there will be delay between the button pressing and the actual movement of the robot car.

Besides having a delay, the movement of the robot is not flexible. This program is made with the turning left and right has a specific-fixed value of angle, and it cannot be changes by the user. The angle is not the perfect one, but at the very least it can give a noticeable movement for right and left turn. And also, when we moved the robot car backward, we cannot do it while turning it left or right, but only in the straight direction. And another error is that the movement of turning left or right will not reset the position of the robot after turning. It causes a little problem when moving it on a narrow road, because it may be hard to control the robot.

## V. Conclusion

To conclude, this project is made to design an automatic wireless mobile weather station. This is a weather station which is put on to the wireless robot car to make it mobile, and thus can be used to go to unreachable location or a dangerous location. This way, it will cause the human no harm, and we can measure the condition at the location without having to go there.

The result is pretty good, since it can be controlled properly and the data can be taken and be read without any problem. The usage is done easily done by one application for two purposes, the robot car controlling and to read the data from the sensors.

Although some improvements can still be implemented, but the basic idea of this project has been achieved.

## REFERENCE

[1] *DC to DC converter,* 2018, [online], Available: https://www.rohm.com/electronics-basics/dc-dc converters/linear-vs- switching-regulators (Accessed on 16th November 2018).

[2] *Controlling DC motor using* Arduino, 2018, [online], Available: https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor- control-tutorial-l298n-pwm-h-bridge/ (Accessed on 16th November 2018).