SMART CAR PARK

Alicia Thum (182004S)

Nicholas Chia (182019C)

EG2010-A3

Project Description

The idea behind this Smart Carpark is to help drivers with parking. There are RGB LED at the front of every parking which will either show green or red. Green signifies that the parking lot is available while red signifies that the parking lot is unavailable. There is also a Grove LCD RGB Backlight at the front of the car park area which will show the amount of lots available within the carpark at every point of time.

Hardware

Transmitter	**Receiver**
Arduino UNO	Arduino UNO
LDR (VT900) x3	RGB LED x3
10kΩ resistor x3	Grove LCD RGB Backlight x1
Jumper wires	270Ω resistor x3
	Jumper wires

Software

1. Grove LCD RGB Backlight

```
#include "rgb_lcd.h"

#include <Wire.h>

int counter = 3;

int state1 = 1;

int state2 = 1;

int state3 = 1

byte message[VW_MAX_MESSAGE_LEN];

byte messageLength = VW_MAX_MESSAGE_LEN;


void setup {

Serial.begin(9600);

Serial.println("Device is ready");


rgb_lcd lcd;


 {

   // set up the LCD's number of columns and rows:

   lcd.begin(16, 2);

   // Print a message to the LCD.
```

```
    lcd.print("carpark: ");
    lcd.print(counter);
    }
}

void loop () {
 Serial.println();
 // Turn off the display:
 lcd.clear();
 counter = state1 + state2 + state3;

    lcd.print("carpark: ");
    lcd.print(counter);
    delay(500);
 }

 Serial.println();
  // Turn off the display:
   lcd.clear();
  counter = state1 + state2 + state3;

    lcd.print("carpark: ");
    lcd.print(counter);
    delay(500);
```

```
}
```

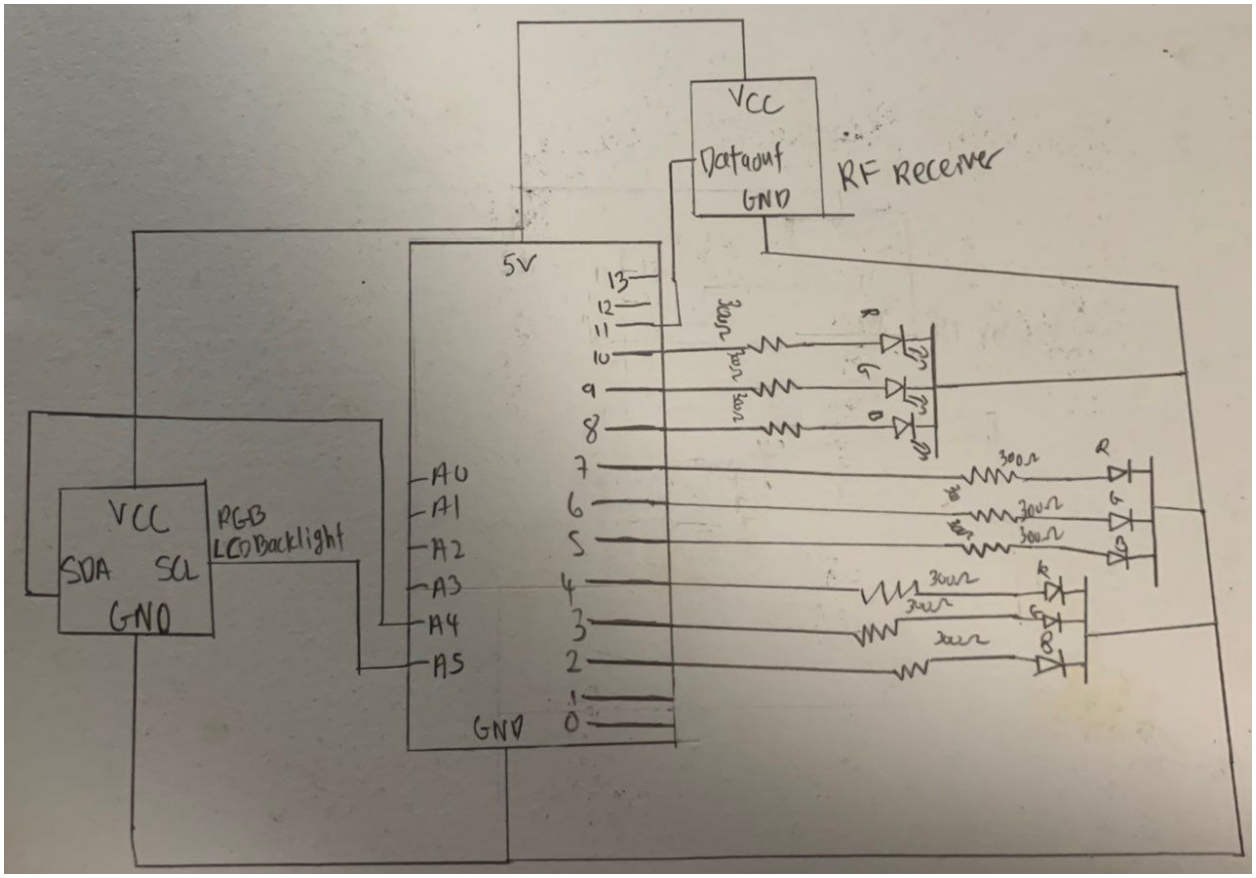
2. RGB LED

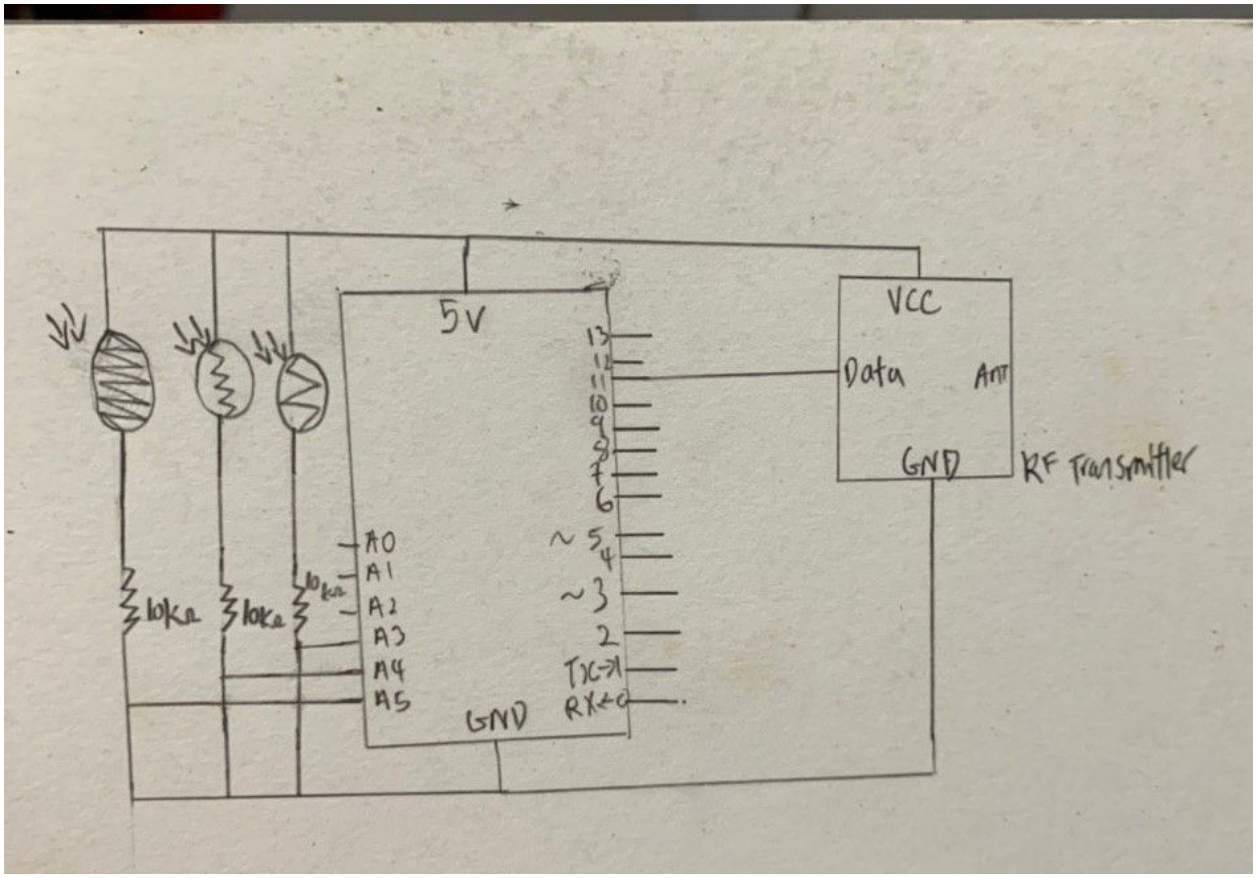
```
int redPin1 = 8;

int greenPin1 = 10;

int bluePin1 = 9;


int redPin2 = 5;

int greenPin2 = 6;

int bluePin2 = 7;


int redPin3 = 2;

int greenPin3 = 3;

int bluePin3 = 4;


void setup {

pinMode(redPin1, OUTPUT);

pinMode(greenPin1, OUTPUT)

pinMode (bluePin1, OUTPUT);


setColor1(0, 0, 255);  // green

  //--------------------------- A2 ------------------------------------------------------------------
```

```
pinMode(redPin2, OUTPUT);
pinMode(greenPin2, OUTPUT);
pinMode (bluePin2, OUTPUT);
setColor2(0, 0, 255);  // green

  //--------------------------- A3----------------------------------------------------------------
pinMode(redPin3, OUTPUT);
pinMode(greenPin3, OUTPUT);
pinMode (bluePin3, OUTPUT);

setColor3(0, 0, 255);  // green
}

void setColor1(int red, int green, int blue)
{
  #ifdef COMMON_ANODE
    red = 255 - red;
    green = 255 - green;
    blue = 255 - blue;
  #endif
  analogWrite(redPin1, red);
  analogWrite(greenPin1, green);
  analogWrite(bluePin1, blue);
}
```

```
void setColor2(int red, int green, int blue)
{

  #ifdef COMMON_ANODE
    red = 255 - red;
    green = 255 - green;
    blue = 255 - blue;
  #endif
  analogWrite(redPin2, red);
  analogWrite(greenPin2, green);
  analogWrite(bluePin2, blue);
}

void setColor3(int red, int green, int blue)
{
#ifdef COMMON_ANODE
    red = 255 - red;
    green = 255 - green;
    blue = 255 - blue;
  #endif
  analogWrite(redPin3, red);
  analogWrite(greenPin3, green);
  analogWrite(bluePin3, blue);
}
```

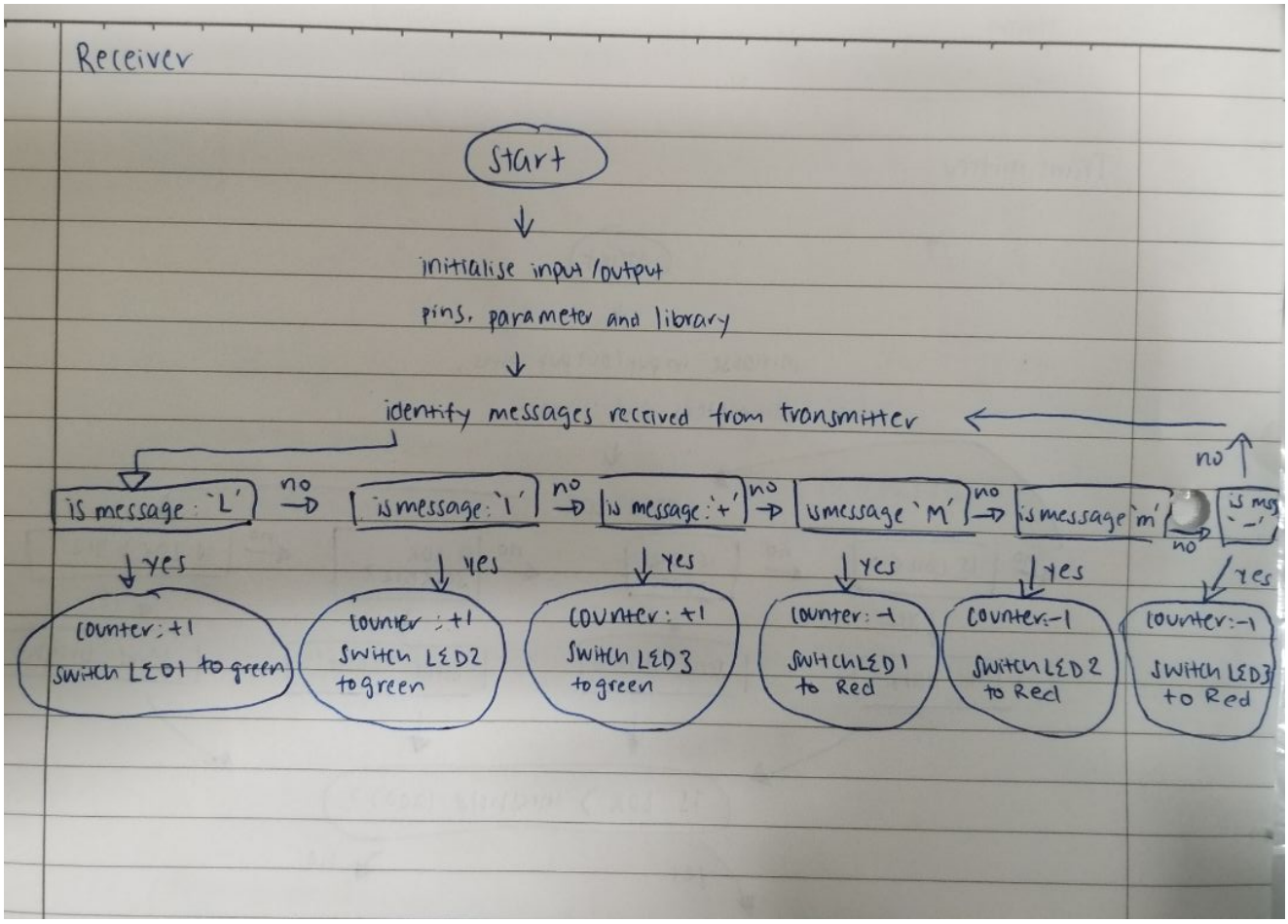
Circuit Diagram



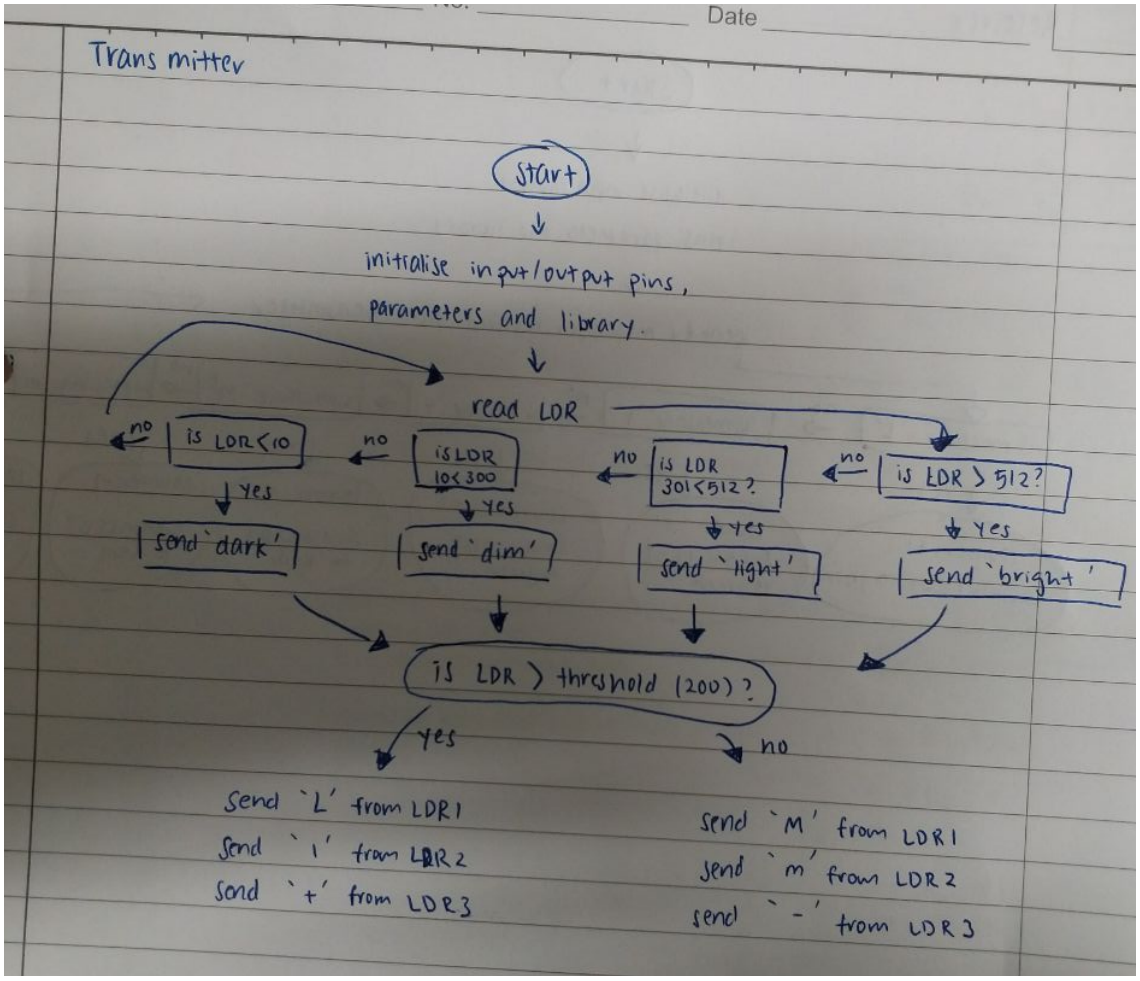

5V
VCC
Data
Ant
GND
RF Transmitter
10kΩ
10kΩ
10kΩ
A0
A1
A2
A3
A4
A5
GND
13
12
11
10
9
8
7
6
~5
4
~3
2
TX→1
RX←0

Flow Chart

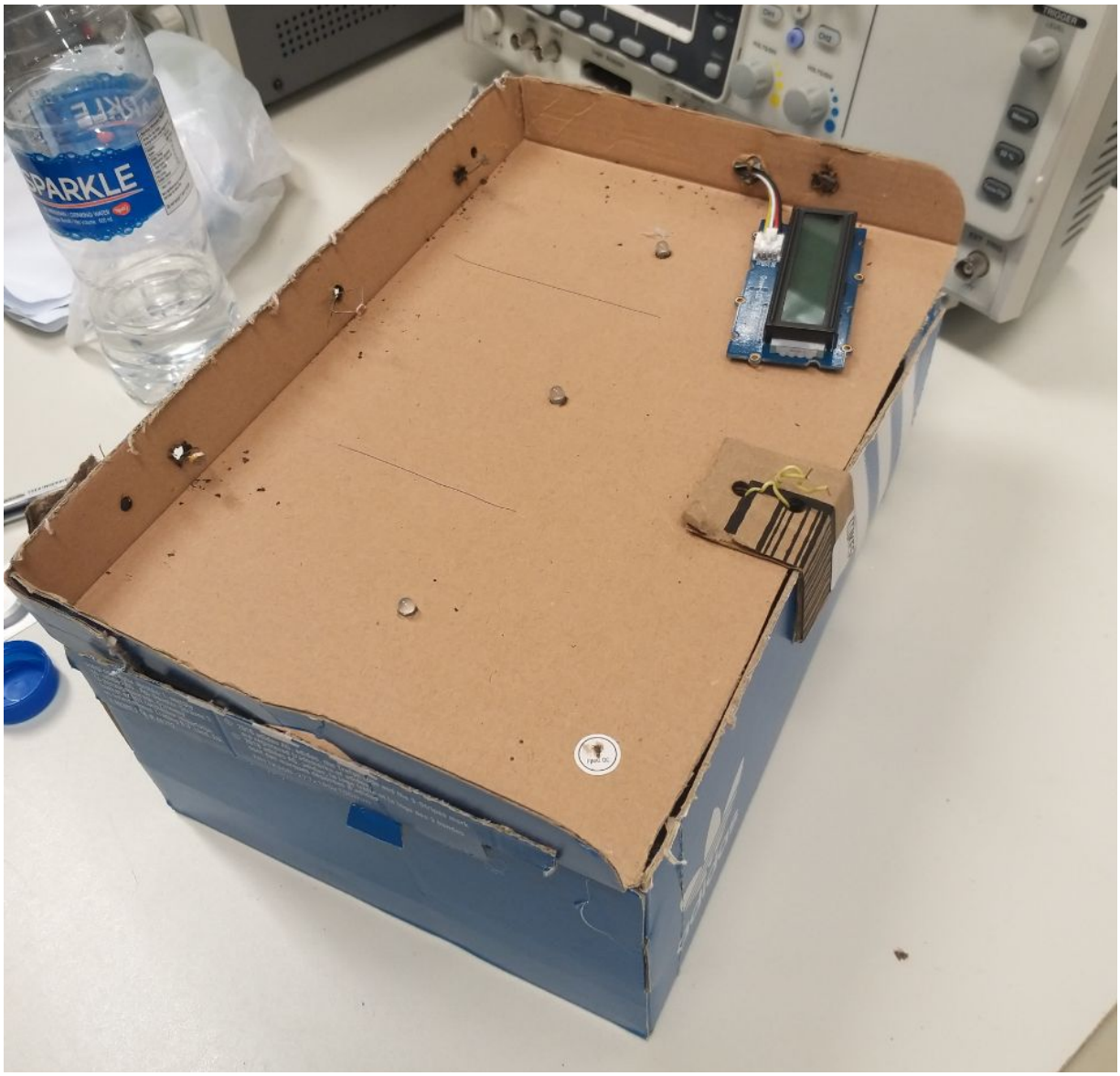
Receiver



Transmitter



Final Product



References

RGB LED - https://learn.adafruit.com/adafruit-arduino-lesson-3-rgb-leds?view=all

Grove LCD RGB Backlight -

https://github.com/Seeed-Studio/Grove_LCD_RGB_Backlight/archive/master.zip

http://wiki.seeedstudio.com/How_to_install_Arduino_Library/

Source Codes (Tx)

```
#include <VirtualWire.h>

//define constant for the pin the sensor is connected to

int ldrPin = A4;

int ldrPin2 = A3;

int ldrPin3 = A2;


// threshold LDR sensor value

const float ldrThreshold = 200;


void setup()

{

  // Initialize the IO and ISR

 vw_setup(2000); // Bits per sec


  // Initialize the serial monitor

  Serial.begin(9600);

}

void loop()

{

  //read the value at A1 and store it in a variable
```

```
int ldrVal = analogRead(ldrPin);


//send the 10-bit sensor value to the serial port
Serial.print("LDR1 value = ");
Serial.print(ldrVal);


if (ldrVal <10)
 {
   Serial.println(", Dark ");
 }


else if (ldrVal <300)
 {
   Serial.println(", Dim ");
 }


else if (ldrVal <512)
 {
   Serial.println(", Light ");
 }


 else
  {
   Serial.println(", Bright ");
```

```
    }
  delay(1000);

  if (ldrVal < ldrThreshold)
   {
    send("L");
    delay(1000);
   }

  else if (ldrVal >= ldrThreshold)
   {
     send("M");
    delay(1000);
   }

//---------------------------------- A2 ----------------------------------------------------------

  //read the value at A2 and store it in a variable
  int ldr2Val = analogRead(ldrPin2);

  //send the 10-bit sensor value to the serial port
  Serial.print("LDR2 value = ");
  Serial.print(ldr2Val);

  if (ldr2Val <10)
```

```
  {
    Serial.println(", Dark ");
  }
else if (ldr2Val <300)
  {
    Serial.println(", Dim ");
  }
else if (ldr2Val <512)
  {
    Serial.println(", Light ");
  }
else
  {
    Serial.println(", Bright ");
  }
delay(1000);
if (ldr2Val < ldrThreshold)
  {
    send("l");
    delay(1000);
  }
  else if (ldr2Val >= ldrThreshold)
    {
      send("m");
```

```
      delay(1000);
    }
//---------------------------------- A3 ----------------------------------------------------------
  //read the value at A3 and store it in a variable
  int ldr3Val = analogRead(ldrPin3);

  //send the 10-bit sensor value to the serial port
  Serial.print("LDR3 value = ");
  Serial.print(ldr3Val);

  if (ldr3Val <10)
   {
     Serial.println(", Dark ");
   }

  else if (ldr3Val <300)
   {
     Serial.println(", Dim ");
  }

  else if (ldr3Val <512)
   {
     Serial.println(", Light ");
   }
```

```
  else
    {
      Serial.println(", Bright ");
    }
  delay(1000);

  if (ldr3Val < ldrThreshold)
  {
    send("+");
    delay(1000);
  }
  else if (ldr3Val >= ldrThreshold)
  {
    send("-");
    delay(1000);
  }
}

void send (char *message)
{
  vw_send((uint8_t *)message, strlen(message));
  // Wait until the whole message is transmitted
  vw_wait_tx();
}
```

Source Codes (Rx)

```
#include "rgb_lcd.h"

#include <Wire.h>


#include <VirtualWire.h>

#include <ServoTimer2.h>


int redPin1 = 8;

int greenPin1 = 10;

int bluePin1 = 9;


int redPin2 = 5;

int greenPin2 = 6;

int bluePin2 = 7;


int redPin3 = 2;

int greenPin3 = 3;

int bluePin3 = 4;


int counter = 3;

int state1 = 1;
```

```
int state2 = 1;
int state3 = 1;

byte message[VW_MAX_MESSAGE_LEN];
byte messageLength = VW_MAX_MESSAGE_LEN;

rgb_lcd lcd;
//uncomment this line if using a Common Anode LED
//#define COMMON_ANODE

void setup()
{
  Serial.begin(9600);
  Serial.println("Device is ready");

  vw_setup(2000); // Bits per sec
  vw_rx_start(); // Start the receiver

  pinMode(redPin1, OUTPUT);
  pinMode(greenPin1, OUTPUT);
  pinMode (bluePin1, OUTPUT);

  setColor1(0, 0, 255);  // green
```

```
  //--------------------------- A2 ------------------------------------------------------------------
 pinMode(redPin2, OUTPUT);
 pinMode(greenPin2, OUTPUT);
 pinMode (bluePin2, OUTPUT);
 setColor2(0, 0, 255);  // green
  //--------------------------- A3 ------------------------------------------------------------------
 pinMode(redPin3, OUTPUT);
 pinMode(greenPin3, OUTPUT);
 pinMode (bluePin3, OUTPUT);
 setColor3(0, 0, 255);  // green

  {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("carpark: ");
  lcd.print(counter);
  }
}

void loop()
{
 if (vw_get_message(message, &messageLength))
 {
```

```
Serial.print("Received: ");
for (int i = 0; i < messageLength; i++)
{
  Serial.write(message[i]);
  if(message[i]=='L')
  {
    state1=0;
    Serial.write(", setColor1(255, 0, 0);  // red");
     setColor1(255, 0, 0);  // red
      delay(1000);
  }

  if(message[i]=='M')
  {
     state1=1;
    Serial.write(", setColor1(0, 0, 255);  // green");
     setColor1(0, 0, 255);  // green
     delay(1000);
  }

  Serial.write(message[i]);
  if(message[i]=='I')
  {
        state2 = 0;
```

```
//  digitalWrite(redPin2, HIGH );
  Serial.write(", setColor2(255, 0, 0);  // red");
   setColor2(255, 0, 0);  // red
    delay(1000);
}
if(message[i]=='m')
{
 state2 = 1;
//  digitalWrite(greenPin2,LOW);
  Serial.write(", setColor2(0, 0, 255);  // green");
  setColor2(0, 0, 255);  // green
   delay(1000);
}

Serial.write(message[i]);
if(message[i]=='+')
{
 state3 = 0;
 // digitalWrite(redPin3, HIGH );
  Serial.write(", setColor3(255, 0, 0);  // red");
   setColor3(255, 0, 0);  // red
    delay(1000);
}
```

```
      if(message[i]=='-')
      {
        state3 = 1;
        //digitalWrite(greenPin3,LOW);
        Serial.write(", setColor3(0, 0, 255);  // green");
         setColor3(0, 0, 255);  // green
          delay(1000);
      }

    }
    Serial.println();
       // Turn off the display:
      lcd.clear();

  counter = state1 + state2 + state3;



    lcd.print("carpark: ");
    lcd.print(counter);
    delay(500);
  }
  }

void setColor1(int red, int green, int blue)
```

```
{
  #ifdef COMMON_ANODE
    red = 255 - red;
    green = 255 - green;
    blue = 255 - blue;
  #endif
  analogWrite(redPin1, red);
  analogWrite(greenPin1, green);
  analogWrite(bluePin1, blue);
}

void setColor2(int red, int green, int blue)
{
  #ifdef COMMON_ANODE
    red = 255 - red;
    green = 255 - green;
    blue = 255 - blue;
  #endif
  analogWrite(redPin2, red);
  analogWrite(greenPin2, green);
  analogWrite(bluePin2, blue);
}

void setColor3(int red, int green, int blue)
```

```
{
  #ifdef COMMON_ANODE
    red = 255 - red;
    green = 255 - green;
    blue = 255 - blue;
  #endif
  analogWrite(redPin3, red);
  analogWrite(greenPin3, green);
  analogWrite(bluePin3, blue);
}
```