

```

#define LED_PIN1 2
#define LED_PIN2 17
#define SPEAKER_PIN 16

#define SDA 13          //Define SDA pins
#define SCL 14          //Define SCL pins

LiquidCrystal_I2C lcd(0x27,16,2);
DHT dht(DHT_PIN, DHT_TYPE);

void sendSensorData() {
  // Ultrasonic sensor
  long duration, distance;
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  duration = pulseIn(ECHO_PIN, HIGH);
  distance = (duration / 2) / 29.1; // Convert to cm

  // DHT sensor
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();

  // Send data to Blynk
  Blynk.virtualWrite(V2, distance);
  Blynk.virtualWrite(V0, temperature);
  Blynk.virtualWrite(V3, humidity);

  // Check if bin is full
  if (distance < 10) {
    Blynk.virtualWrite(V1, "Waste bin is full! ");
    //tone(16, 262, 250);
    tone(SPEAKER_PIN,262,250);
    digitalWrite(LED_PIN2, HIGH);
    digitalWrite(LED_PIN1, LOW);
    lcd.clear();
    lcd.print("Bin is Full!");
  }
}

```

```

}
else{
  Blynk.virtualWrite(V1,"Bin has Space.");
  digitalWrite(LED_PIN1, HIGH);
  digitalWrite(LED_PIN2, LOW);
  lcd.clear();
  lcd.print("Bin has Space");
}
delay(5000);
}

```

BlynkTimer timer;

// This function is called every time the Virtual Pin 0 state changes

BLYNK_WRITE(V4)

```

{
  Serial.println("Inside Blynk Write");
  if(param.asInt() == 1)
  {
    Serial.println("Blynk Write: Value is 1");
    //digitalWrite(2, HIGH);
    digitalWrite(LED_PIN1, HIGH);
    Blynk.virtualWrite(V1,"Started Successfully.");
    sendSensorData();
  }
  else
  {
    Serial.println("Blynk Write: Value is 0");
    //digitalWrite(2, LOW);
    digitalWrite(LED_PIN1, LOW);
    digitalWrite(LED_PIN2, HIGH);
  }
}

```

// This function is called every time the device is connected to the Blynk.Cloud

BLYNK_CONNECTED()

```

{
  Blynk.syncVirtual(V0);
  Blynk.syncVirtual(V1);
}

```

```

Blynk.syncVirtual(V2);
Blynk.syncVirtual(V3);
Blynk.syncVirtual(V4);
Serial.println("Inside Blynk: Blynk is Connected");
lcd.clear();
lcd.print("Blynk -Connected");
delay(5000);
}

```

// This function sends Arduino's uptime every second to Virtual Pin 2.

```

void myTimerEvent()
{
  Blynk.virtualWrite(V0, millis() / 1000);
  Blynk.virtualWrite(V1, millis() / 1000);
  Blynk.virtualWrite(V2, millis() / 1000);
  Blynk.virtualWrite(V3, millis() / 1000);
  Blynk.virtualWrite(V4, millis() / 1000);
}

```

```

bool i2CAddrTest(uint8_t addr) {
  Wire.begin();
  Wire.beginTransmission(addr);
  if (Wire.endTransmission() == 0) {
    return true;
  }
  return false;
}

```

```

void setup()
{
  Wire.begin(SDA, SCL);
  if (!i2CAddrTest(0x27)) {
    lcd = LiquidCrystal_I2C(0x3F, 16, 2);
  }
  lcd.init();
  lcd.backlight();

  lcd.setCursor(0,0);
  lcd.print("Smart Campus");
}

```

```
lcd.setCursor(0,1);  
lcd.print("Waste Management");  
//delay(2500);  
//lcd.clear();  
lcd.setCursor(0,2);  
lcd.print(" Version 0.0.1");  
//
```

```
pinMode(5, INPUT_PULLUP);  
//pinMode(2, OUTPUT);  
//pinMode(17, OUTPUT);  
pinMode(LED_PIN1, OUTPUT);  
pinMode(LED_PIN2, OUTPUT);  
pinMode(SPEAKER_PIN, OUTPUT);  
// Begin.  
Serial.begin(115200);  
Serial.println("Serial Prints Starts...");
```

```
Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
```

```
pinMode(TRIG_PIN, OUTPUT);  
pinMode(ECHO_PIN, INPUT);
```

```
dht.begin();
```

```
// Set a timer to send sensor data every 10 seconds  
timer.setInterval(10000L, sendSensorData);  
}
```

```
void loop()
```

```
{  
  Blynk.run();  
  timer.run();
```

```
// DHT sensor
```

```
float temperature = dht.readTemperature();  
float humidity = dht.readHumidity();
```

```
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("Temp: " + String(temperature, 2) + " C");  
lcd.setCursor(0,1);  
lcd.print("Humidity: " + String(humidity, 1) + "%");  
  
//lcd.setCursor(0,1);  
//lcd.print("Counter:");  
//lcd.print(millis() / 1000);  
delay(5000);  
}
```