

1. Sensor Modules

a. Temperature and Humidity Sensor:

- **Module:** DHT11 or DHT22
- **Library:** [DHT.h](#)
- **Purpose:** Measures temperature and humidity.
- **Example Code:**

```
#include <DHT.h>

#define DHTPIN 2
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

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

void loop() {
  float temperature = dht.readTemperature();
  float humidity = dht.readHumidity();
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.print(" °C, Humidity: ");
  Serial.print(humidity);
  Serial.println(" %");
  delay(2000);
}
```

b. Barometric Pressure Sensor:

- **Module:** BMP180 or BMP280
- **Library:** [Adafruit_BMP085_U.h](#) (for BMP180) or [Adafruit_BMP280.h](#) (for BMP280)
- **Purpose:** Measures atmospheric pressure and can be used to estimate altitude.
- **Example Code**

```
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP085_U.h>
```

```
Adafruit_BMP085_Unified bmp = Adafruit_BMP085_Unified();
```

```
void setup() {  
  Serial.begin(9600);  
  if (!bmp.begin()) {  
    Serial.print("Couldn't find the sensor");  
    while (1);  
  }  
}
```

```
void loop() {  
  sensors_event_t event;  
  bmp.getEvent(&event);  
  if (event.pressure) {  
    Serial.print("Pressure: ");  
    Serial.print(event.pressure);  
    Serial.println(" hPa");  
  }  
  delay(2000);  
}
```

c. Load Cell:

- **Module:** HX711
- **Library:** [HX711.h](#)
- **Purpose:** Measures weight using a load cell.
- **Example Code:**

```
#include <HX711.h>
```

```
#define LOADCELL_DOUT_PIN 3  
#define LOADCELL_SCK_PIN 4  
HX711 scale;
```

```
void setup() {  
  Serial.begin(9600);  
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);  
}
```

```
void loop() {  
  float weight = scale.get_units(10);  
  Serial.print("Weight: ");  
  Serial.println(weight);  
  delay(2000);  
}
```

d. Ultrasonic Distance Sensor:

- **Module:** HC-SR04
- **Library:** No specific library required, but you can use simple pulse timing.
- **Purpose:** Measures distance using ultrasonic waves.
- **Example Code**

```
#define TRIG_PIN 9
#define ECHO_PIN 10

void setup() {
  Serial.begin(9600);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
}

void loop() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);

  long duration = pulseIn(ECHO_PIN, HIGH);
  float distance = (duration / 2.0) * 0.0344;

  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(2000);
}
```

2. Connectivity Module

a. Sending Data:

- **Module:** WiFi (e.g., ESP8266) or GSM (e.g., SIM800)
- **Library:**
 - For ESP8266: `ESP8266WiFi.h`
 - For GSM: `SIM800L.h` (or similar based on your module)
- **Purpose:** Sends sensor data to a server or cloud.
- **Example Code for ESP8266:**

```
#include <ESP8266WiFi.h>

const char* ssid = "your_SSID";
const char* password = "your_PASSWORD";
```

```

WiFiClient client;

void setup() {
  Serial.begin(9600);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("Connected to WiFi");
}

void loop() {
  if (client.connect("server_address", port)) {
    client.print("GET /update?value=");
    client.print(sensorValue);
    client.println(" HTTP/1.1");
    client.println("Host: server_address");
    client.println("Connection: close");
    client.println();
  }
  delay(5000);
}

```

3. Controlling Mechanisms

a. Motor and Ejection Charges:

- **Module:** Relay or MOSFET for switching
- **Library:** No specific library required, use `digitalWrite()` to control.
- **Example Code**

```

#define MOTOR_PIN 7
#define CHARGE_PIN 8

void setup() {
  pinMode(MOTOR_PIN, OUTPUT);
  pinMode(CHARGE_PIN, OUTPUT);
}

void loop() {
  // Motor ignition example
  digitalWrite(MOTOR_PIN, HIGH);
  delay(5000); // 5 seconds
  digitalWrite(MOTOR_PIN, LOW);

  // Ejection charge example

```

```

digitalWrite(CHARGE_PIN, HIGH);
delay(2000); // 2 seconds
digitalWrite(CHARGE_PIN, LOW);

delay(10000); // Wait before next cycle
}

```

Integrating All Modules

Here's a simplified example integrating all the modules:

```

#include <DHT.h>
#include <Adafruit_BMP085_U.h>
#include <HX711.h>
#include <ESP8266WiFi.h>

// Define pins and modules
#define DHTPIN 2
#define DHTTYPE DHT22
#define LOADCELL_DOUT_PIN 3
#define LOADCELL_SCK_PIN 4
#define ECHO_PIN 5
#define TRIG_PIN 6
#define MOTOR_PIN 7
#define CHARGE_PIN 8

DHT dht(DHTPIN, DHTTYPE);
Adafruit_BMP085_Unified bmp = Adafruit_BMP085_Unified();
HX711 scale;

const char* ssid = "your_SSID";
const char* password = "your_PASSWORD";
WiFiClient client;

void setup() {
  Serial.begin(9600);
  dht.begin();
  bmp.begin();
  scale.begin(LOADCELL_DOUT_PIN, LOADCELL_SCK_PIN);

  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  pinMode(MOTOR_PIN, OUTPUT);
  pinMode(CHARGE_PIN, OUTPUT);

  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {

```

```

    delay(500);
    Serial.print(".");
}
Serial.println("Connected to WiFi");
}

void loop() {
    // Read sensors
    float temperature = dht.readTemperature();
    float humidity = dht.readHumidity();

    sensors_event_t event;
    bmp.getEvent(&event);
    float pressure = event.pressure;

    float weight = scale.get_units(10);

    // Measure distance
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);
    long duration = pulseIn(ECHO_PIN, HIGH);
    float distance = (duration / 2.0) * 0.0344;

    // Print sensor values
    Serial.print("Temp: "); Serial.print(temperature); Serial.print(" C ");
    Serial.print("Humidity: "); Serial.print(humidity); Serial.print(" % ");
    Serial.print("Pressure: "); Serial.print(pressure); Serial.print(" hPa ");
    Serial.print("Weight: "); Serial.print(weight); Serial.print(" g ");
    Serial.print("Distance: "); Serial.print(distance); Serial.println(" cm");

    // Send data
    if (client.connect("server_address", port)) {
        client.print("GET /update?temp=");
        client.print(temperature);
        client.print("&hum=");
        client.print(humidity);
        client.print("&press=");
        client.print(pressure);
        client.print("&weight=");
        client.print(weight);
        client.print("&dist=");
        client.print(distance);
        client.println(" HTTP/1.1");
        client.println("Host: server_address");
        client.println("Connection: close");
    }
}

```

```
    client.println();  
}  
  
// Example control logic  
if (distance < 10) { // Example condition  
    digitalWrite(MOTOR_PIN, HIGH);  
    delay(5000);  
    digitalWrite(MOTOR_PIN, LOW);  
}  
  
delay(10000); // Wait before next loop  
}
```