Project Deliverable F: Prototype 1 and Customer Feedback

Ayan Aman
Farah Elkelish
Murad Ibrahim
Jérémie Losier
Michael Shokralla

November 4, 2021

Abstract

The main purpose of the project is to start putting the first prototype together. It is important to include client and TA feedback to update and improve the ideas. The CO, temperature and motion sensors are tested on tinkercad and assembled together in one simulation. The arduino codes are revised and they are combined for the assembly of all sensors. There are also testing plans included to verify the compatibility, battery longevity, and bluetooth connectivity.

Table of contents

Introduction

______The prototyping phase is the most important phase for the development of a device. Prototypes can explain the basic concepts of a device and are useful to receive feedback from clients and possible users. A prototype also usually has smaller goals and isn't the same as the development of the fully functional device. The first prototype for the child/pet monitoring device is focused on the hardware system's parts and how those parts interact with each other. There are three sensors: the TMP 36 (temperature sensor), The MQ-7 (carbon monoxide detector) and the PIR motion sensor. Their functionality will first be tested separately using a simulator website called TinkerCad which allows us to virtually make the arduino circuit and add the code as well as run the code and see the results. When the desired results are achieved then the physical circuit can be tested. Secondly, all of the sensors will be tested together to check if they properly interact with one another. This step will be achieved by first writing a full code that incorporates the codes for the individual circuits. Finally, when the prototype is presentented the feedback will be implemented onto the second prototype.

Prototype material list:

- Arduino Uno
- TMP 36
- Breadboard
- Motion Sensor
- Tinkercad
- Jumper Wires
- Thunkable App Builder
- USB 2.0 Cable Type A/B
- Arduino App

Hypothesis: Apparatus must run within the vehicle and detect variations in temperature, CO levels, and motion.

Test 1 Sensors detection testing

We will test out the functionality of the sensors. We will check if the sensors can detect the CO level, the temperature and the presence of the kids based on their motion after being left alone in the car.

1.1 Temperature sensor

Based on a Tinkercad simulation using the setup featured in Figure 1. The TMP 36 does give an accurate reading of the temperature in Celsius. The serial monitor shows "Current Temperature:" before writing the temperature. The degree of precision in the reading of the temperature is in the hundredths decimal.

Code :

```
int sensePin = A0;
int sensorInput;
double temp;

void setup() {

  Serial.begin(9600);
}
void loop() {

  sensorInput = analogRead(A0);
  temp = (double)sensorInput / 1024;
  temp = temp * 5;
  temp = temp - 0.5;
  temp = temp * 100;  // makes into degrees celsius

Serial.print("Current Temperature: ");
Serial.println(temp);

}
```

1.1.1Tinkercad simulation test

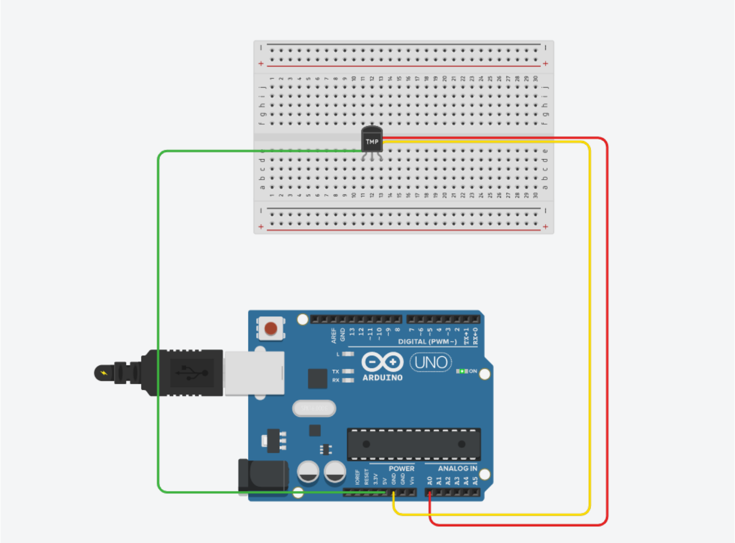


Figure 1: The Tinkercad Simulation setup for the TMP 36

1.1.2 Physical prototype Temperature sensor test

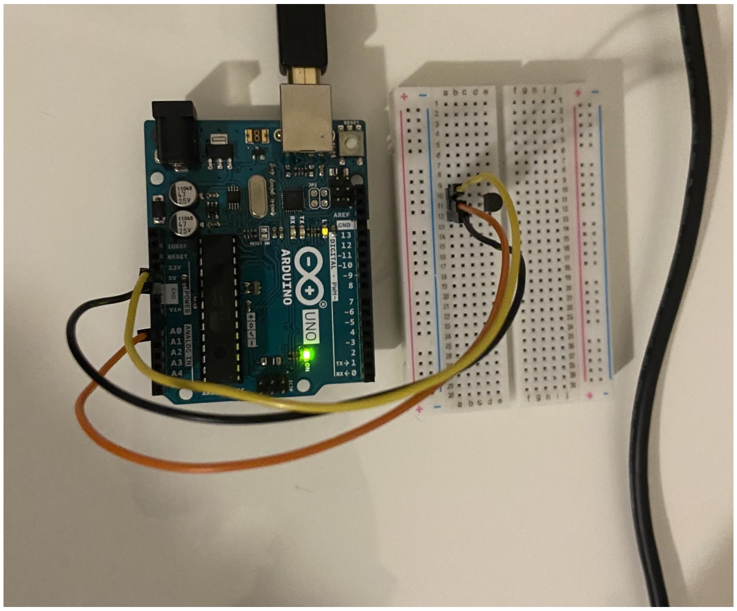
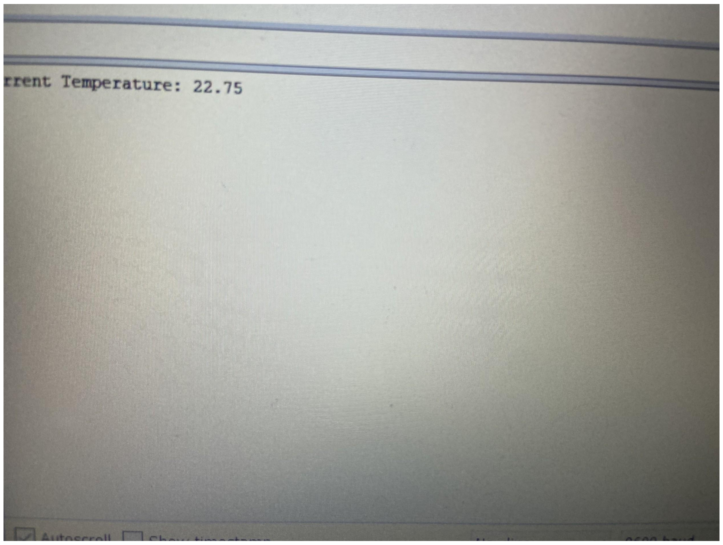


Figure 2: A physical representation of the setup used for the TMP 36



```
rrent Temperature: 22.75
```

Figure 3 : The serial monitor obtained from the physical testing of the TMP 36

1.2 Motion sensor

_______Based on Tinkercad, we were able to recreate Figure 2.All it does is detect the presence of motion in its range of 110 degrees at a delay time of 2000 milliseconds. There were a couple mistakes in the code such as typos and missing brackets. We also added an else if statement mentioning the absence of motion. That's the newly updated code:

```
int pirPin = 2;
int count;

void setup (){
Serial.begin (9600);
pinMode (pirPin, INPUT);
count = 0;
}
void loop (){
int pirVal = digitalRead(pirPin);



if (pirVal==HIGH){
     count++;
  Serial.print("Motion Detected :");
```

```
     Serial.println(count);
  delay (2000); }


 else
   {
     count++;
            Serial.print("Motion Not Detected :");
     Serial.println(count);
     delay (2000); }
}
```

1.2.1 Tinkercad simulation testing

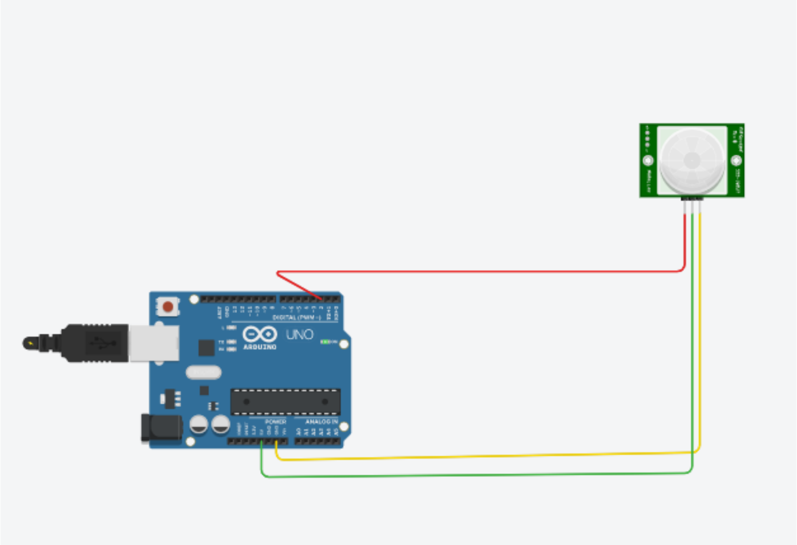


Figure 4: The Tinkercad Simulation setup for the Motion sensor

1.2.2 Physical prototype Motion sensor test

The motion sensor was tested by doing movement in the field of view of the sensor and stopping the movement to test if it detects movement and no movement. The code worked and it can be seen on the serial monitor when there was movement the monitor printed “ Motion detected” and when there was none it printed “ Motion Not Detected”.

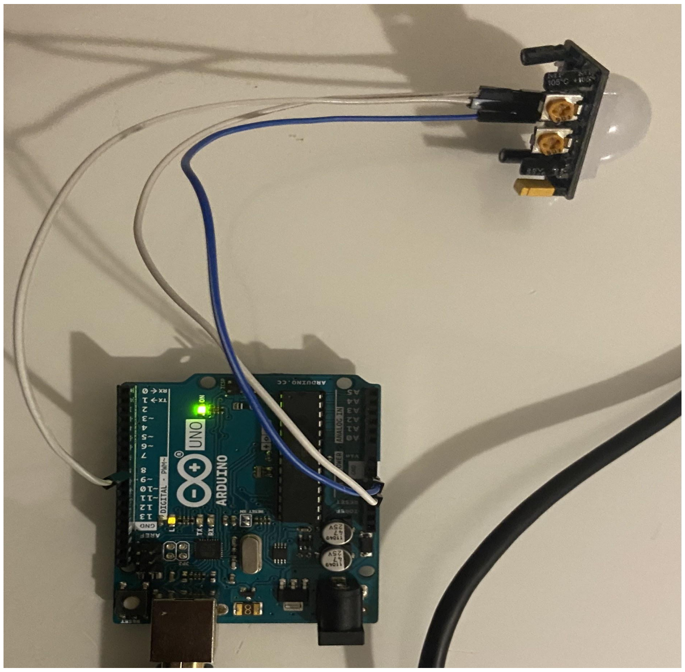
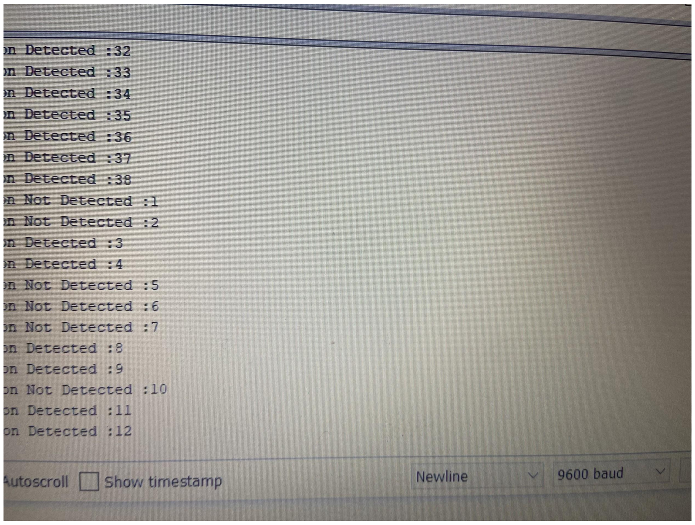


Figure 5: A physical representation of the setup used for the motion sensor



```
on Detected :32
on Detected :33
on Detected :34
on Detected :35
on Detected :36
on Detected :37
on Detected :38
on Not Detected :1
on Not Detected :2
on Detected :3
on Detected :4
on Not Detected :5
on Not Detected :6
on Not Detected :7
on Detected :8
on Detected :9
on Not Detected :10
on Detected :11
on Detected :12
```

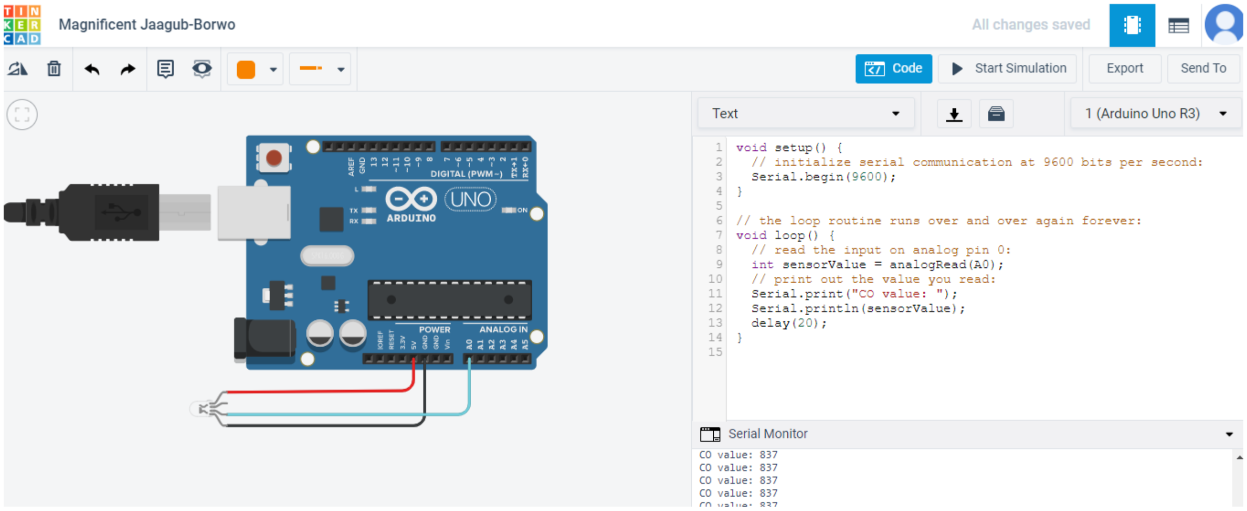
Figure 6: The serial monitor obtained from the physical testing of the Motion sensor

1.3 CO sensor

A LED was used instead of a CO sensor in this simulation as a temporary replacement because the CO sensor is not available on tinkercad. The code was updated in this simulation:

```
void setup() {
  Serial.begin(9600);
}
void loop() {
   int sensorValue = analogRead(A0);
 // prints out the value is read:
 Serial.print("CO value: ");
 Serial.println(sensorValue);
 delay(20);
}
```

1.3.1 Tinkercad simulation test



```
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.print("CO value: ");
  Serial.println(sensorValue);
  delay(20);
}
```

```
CO value: 837
CO value: 837
CO value: 837
CO value: 837
```

Figure 7: The Tinkercad Simulation setup for the CO sensor using a LED light

<u>Test 2, Compatibility verification</u>:

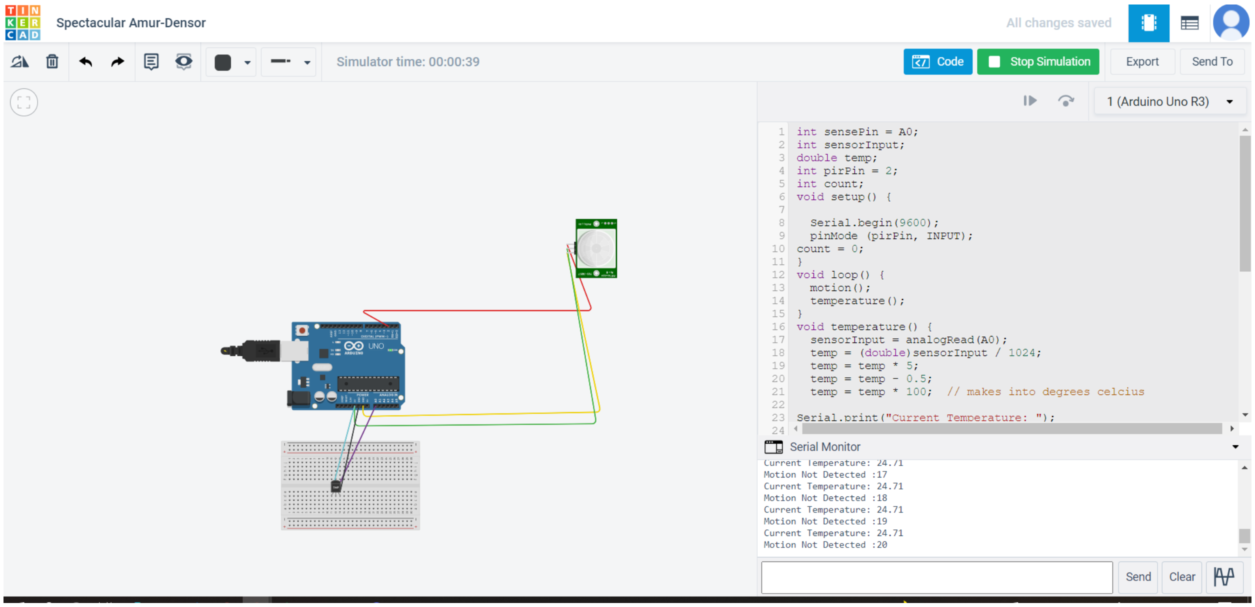
Check to see if all components work/interact with each other properly on initial start-up.

2.1 Code

```
int AOUTpin = A1;
int sensorValue;
int sensePin = A0;
int sensorInput;
double temp;
int pirPin = 2;
int count;
void setup() {

  Serial.begin(9600);
  pinMode (pirPin, INPUT);
count = 0;
}
void loop() {
  motion();
  temperature();
 carbonmonoxide();
}
void temperature() {
  sensorInput = analogRead(A0);
  temp = (double)sensorInput / 1024;
  temp = temp * 5;
  temp = temp - 0.5;
  temp = temp * 100;  // makes into degrees celsius

Serial.print("Current Temperature: ");
Serial.println(temp);


}
```

```
// motion
void motion (){
int pirVal = digitalRead(pirPin);



if (pirVal==HIGH){
      count++;
  Serial.print("Motion Detected :");
      Serial.println(count);
  delay (2000); }


 else
   {
      count++;
               Serial.print("Motion Not Detected :");
      Serial.println(count);
      delay (2000); }
}
void carbonmonoxide(){

      sensorValue=analogRead(AOUTpin) ;

      Serial.println(sensorValue);

      delay(1000); //for stability

}
```

2.2 Tinkercad simulation screenshot



```
int sensePin = A0;
int sensorInput;
double temp;
int pirPin = 2;
int count;
void setup() {

  Serial.begin(9600);
  pinMode (pirPin, INPUT);
count = 0;
}
void loop() {
  motion();
  temperature();
}
void temperature() {
  sensorInput = analogRead(A0);
  temp = (double)sensorInput / 1024;
  temp = temp * 5;
  temp = temp - 0.5;
  temp = temp * 100;  // makes into degrees celcius

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

```
Current Temperature: 24.71
Motion Not Detected :17
Current Temperature: 24.71
Motion Not Detected :18
Current Temperature: 24.71
Motion Not Detected :19
Current Temperature: 24.71
Motion Not Detected :20
```

Figure 8: The Tinkercad Simulation setup for the TMP 36 and the motion sensor

2.3 Physical compatibility testing

To test both the motion sensor and the temperature sensor both of the sensors will be put in parallel on the breadboard.

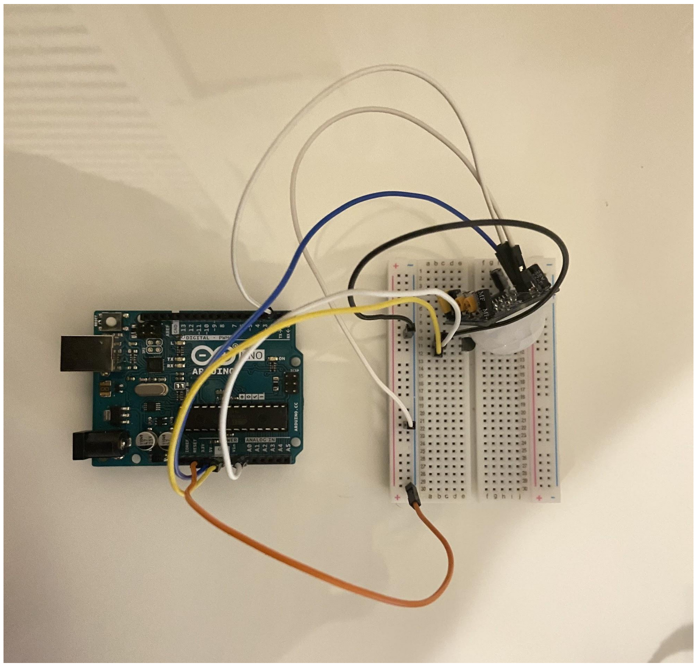
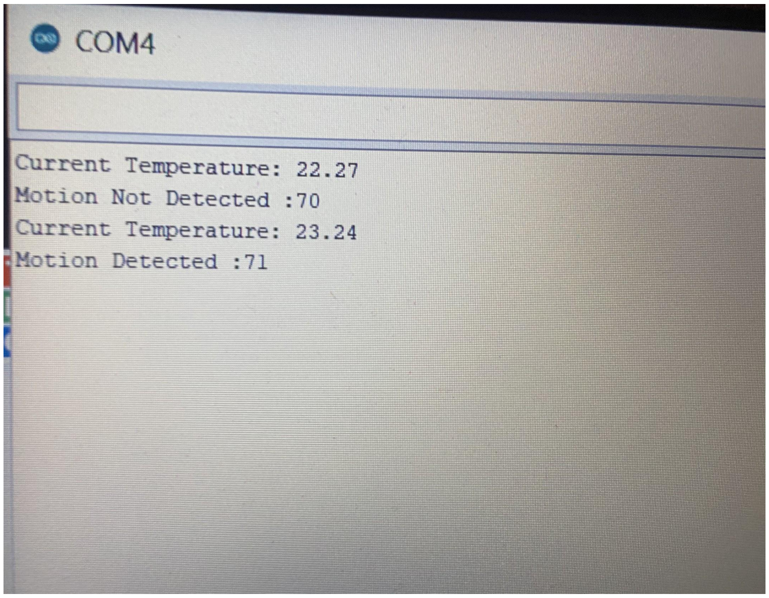


Figure 9: A physical representation of the setup used for the motion sensor and the TMP 36 collectively



```
Current Temperature: 22.27
Motion Not Detected :70
Current Temperature: 23.24
Motion Detected :71
```

Figure 10 : The serial monitor obtained from the physical testing of the Motion sensor and the TMP 36 collectively

***Test 3 Bluetooth connectivity*:**

We will run the device and send some signals through bluetooth to make sure the mobile app gets the notifications properly and there are no flaws in this functionality since it is a life or death situation.We will test these functionalities by sending various strings of data in the arduino . The data will go from:

- Arduino code
- HC-05
- Bluetooth
- Cellphone code
- Display message on phone

If everything goes well the computer should be able to send a message to the user before he or she gets away from his car

Feedback Implementation:

- *PIR sensor to be changed places*
- *9V batteries instead of lipo batteries*
- *Do research on CO sensor*

- *Do research on bluetooth connectivity*
- *Do project on android studio*
- *Build a case*

Prototype 2 test plan:

Test 1: Case Fitting Test

The children shouldn't be in contact with electrical components of the system. Therefore, a case will be built fitting these components together in one spot. The case should be durable and big enough to fit the electrical components.

Test 2: Power Provision Test

The batteries should be able to provide enough voltage to empower all the sensors and the Arduino. The device must also be able to stand on its own running on a single 9v battery

Test 3: App Simplicity Test

The phone application should be easily maintained by the driver. In order to do that, the app needs to be simple enough for them to understand the information provided from it.

After 3 main tests, obtain some valuable consumer input by employing some testers for the product. Gather the results in tabulated form, and analyze the results for potential improvements.

Conclusion:

To conclude, a physical prototype was built using the materials at our disposal : Arduino Uno , breadboard , motion sensor, temperature sensor and some jumper wires. Using the Tinkercad simulation, the codes for the motion, the temperature and the CO sensors were tested ,individually and collectively, and corrected if needed. Then, a physical testing using the correct coding occurred with the available sensors. Finally, some feedback was collected to improve the next prototype and a test plan for it was established.