

Prototype II and Customer Feedback

Team Members: Pthahnil Guo, Zijie Chen, Paulo Meneses, Cisco Musetti, Ben Paul

INTRODUCTION:

In this deliverable, we will devise a testing plan, develop our second prototype, and document our testing processing based on our first prototype. In addition, the explanations of how previous results from prototype I continue in this prototype, analytic analysis of this prototype, and how will our team gather and apply the feedbacks for next client meeting will also include. Furthermore, the stopping criteria for testing and acceptable fidelities

PROTOTYPING TEST PLAN:

1) Why are we doing this test:

After we identify and prove the positions of several sensors of our conceptual design in prototype I, the next step is to compose and verify the codes of different types of sensors: detective, temperature, and infrared. In other words, prototype II is the steps that ensures our integrated sensors will be functional, so we can move to prototype III to modify the coding of sensors on Bowie in order to reach our expectation and solve the problem. Furthermore, we do want to use this prototype as the tool to verify the feasibility of our design, and to perform analytical analysis to reduce risk and uncertainty for next prototype.

2) Test Objectives Description:

A. Specific test objectives:

- 1) Verifying the feasibility of coding for different types of sensors, as well as to determine whether the codes will make sensors functional.
- 2) Verifying the feasibility of circuit connections between Arduino and different sensors.
- 3) Analyze the potential problems and risk of failure of the coding.
- 4) Explore new integrated ideas that improve the efficiency of the sensors, and acts as the communication tools for next client meeting.

B. What being learned & communicated with the prototype:

The coding for both detective and temperature sensors have been verified correctly by Arduino programming. Although the coding for detective sensors is correct, we still need to improve its effectiveness and come up an integrated coding to make sure two detective sensors can work together. Furthermore, our team will also work on discovering the improvements for coding and sensors' precision.

C. Possible types of result:

There are mainly too kind of possible results. First is our team successfully reach the objectives of prototype II, that circuit connections for Arduino and coding for different types of sensor are both functional and working will together. Second situation is we may fail the circuit connections for Arduino. Because our team has almost finished the program of sensors, the circuit connections is the only left task and our team lack members who have strong skill at this.

D. How will results be used to make decisions:

In first situation, we will present our prototype and testing results to communicate with our client in next meeting with the purpose of seeking for improvements and feedbacks about next prototype. In detail, we will ask the acceptable and idea values, such as detective distance and detection period, for the sensors that client desired for final prototype. Then, we will modify the prototype base on feedback. In second situation, we will complete our circuit connections during next lab section with the helps from our TAs.

E. Criteria for test success or failure:

The success or failure of the coding is mainly depending on the system checking in Arduino programming software. For circuit connection, the stopping criteria is whether the sensors are functional according to serial monitor after they are connected to Arduino board, this is also the other criteria for sensors' coding.

3) What is going on and how is it being done:

A. The reason for selection of this type of prototype:

Our team choose the type of comprehensive analytical prototype as our second prototype, because we are working on the coding for each type of sensor during this stage and it is important to ensure that all sensors will be functional when they are connected to Arduino board. In other words, this prototype is mainly focus on the coding of an integrated system and the analysis of the coding and the system, so the comprehensive analytical prototype is one of most suitable types.

B. Testing process:

We verify the codes of different type of sensors by Arduino IDE firstly, and then if the codes are correct in put the codes in to Arduino Uno board and open the serial monitor to prove whether the sensors working.

1) Infrared sensor:

a) Code:

```

r_code | Arduino 1.8.7 (Windows Store 1.8.15.0)
File Edit Sketch Tools Help

r_code $
#include <irremote.h>

int led = 13;           // the pin that the LED is attached to
int sensor = 2;        // the pin that the sensor is attached to
int state = LOW;       // by default, no motion detected
int val = 0;           // variable to store the sensor status (value)

void setup() {
  pinMode(led, OUTPUT); // initialize LED as an output
  pinMode(sensor, INPUT); // initialize sensor as an input
  Serial.begin(9600); // initialize serial
}

void loop() {
  val = digitalRead(sensor); // read sensor value
  if (val == HIGH) { // check if the sensor is HIGH
    digitalWrite(led, HIGH); // turn LED ON
    delay(100); // delay 100 milliseconds

    if (state == LOW) {
      Serial.println("Motion detected!");
      state = HIGH; // update variable state to HIGH
    }
  }
  else {
    digitalWrite(led, LOW); // turn LED OFF
    delay(100); // delay 100 milliseconds
    if (state == HIGH) {
      Serial.println("Motion stopped!");
      state = LOW; // update variable state to LOW
    }
  }
}

```

Done uploading
 Sketch uses 2662 bytes (8%) of program storage space. Maximum is 32256 bytes.
 Global variables use 432 bytes (21%) of dynamic memory, leaving 1616 bytes for local variables. Maximum is 2048 bytes.

28 Arduino/Genuino Uno on COM3

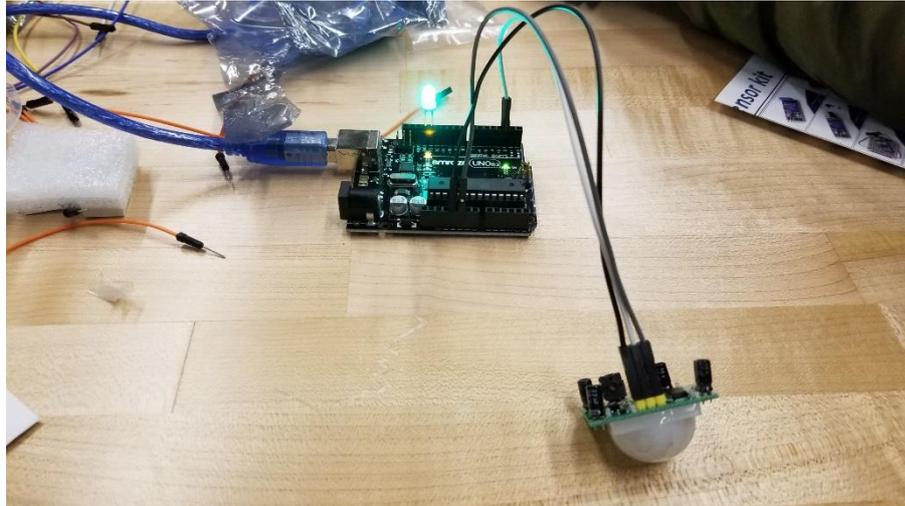
Code has been checked correctly by Arduino IDE.

b) Serial monitor:

The monitor shown the whether the motions have been detected by infrared sensors, which proved the sensor is functional.

```
COM3 (Arduino/Genuino Uno)
Motion detected!
Motion stopped!
Motion detected!
```

c) Actual look of testing



2) Detective sensor:

a) Code:

```
distance_sensor
int trigPin = 11; // Trigger
int echoPin = 12; // Echo
long duration, cm, inches;

void setup() {
  //Serial Port begin
  Serial.begin(9600);
  //Define inputs and outputs
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  // The sensor is triggered by a HIGH pulse of 10 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Read the signal from the sensor: a HIGH pulse whose
  // duration is the time (in microseconds) from the sending
  // of the ping to the reception of its echo off of an object.
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);

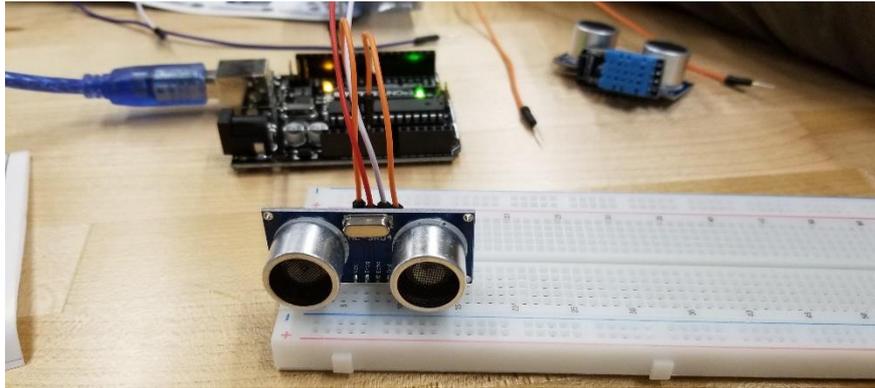
  // Convert the time into a distance
  cm = (duration/2) / 29.1; // Divide by 29.1 or multiply by 0.0343
  inches = (duration/2) / 74; // Divide by 74 or multiply by 0.0135

  Serial.println(inches);
  Serial.println("in, ");
  Serial.println(cm);
  Serial.println("cm");
  Serial.println("");
  delay(250);
}
```

b) Serial monitor:

```
COM3 (Arduino/Genuino Uno)
60in, 154cm
51in, 130cm
62in, 158cm
61in, 156cm
57in, 146cm
50in, 128cm
45in, 115cm
50in, 127cm
33in, 84cm
```

c) Actual look of testing:



3) Temperature and humidity sensor:

a) Code:

```
Project | Arduino 1.8.7 (Windows Store 1.8.15.0)
File Edit Sketch Tools Help

Project $

// For DHT11,
// VCC: 5V or 3V
// GND: GND
// DATA: 2
int pinDHT11 = 2;
SimpleDHT11 dht11;

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

void loop() {
  // Start working...
  Serial.print("Temperature and Humidity");

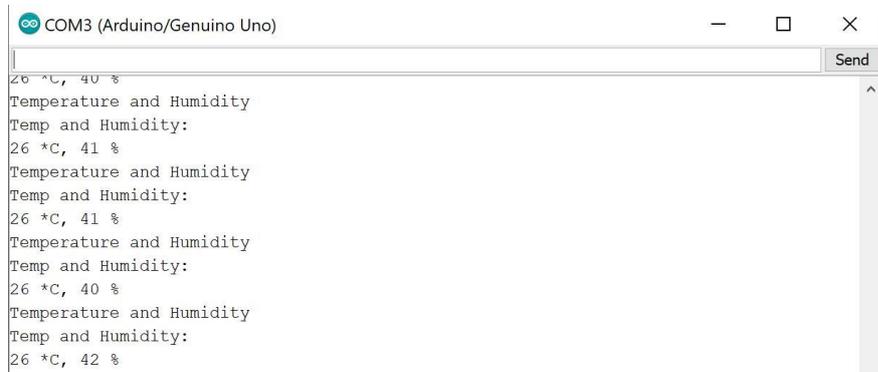
  // read with raw sample data.
  byte temperature = 0;
  byte humidity = 0;
  byte data[4] = {0};
  if (dht11.read(pinDHT11, &temperature, &humidity, data) {
    Serial.print("Read DHT11 failed");
    return;
  }

  Serial.println("");
  Serial.println("Temp and Humidity:");
  Serial.print((int)temperature); Serial.print(" *C, ");
  Serial.print((int)humidity); Serial.println(" %");

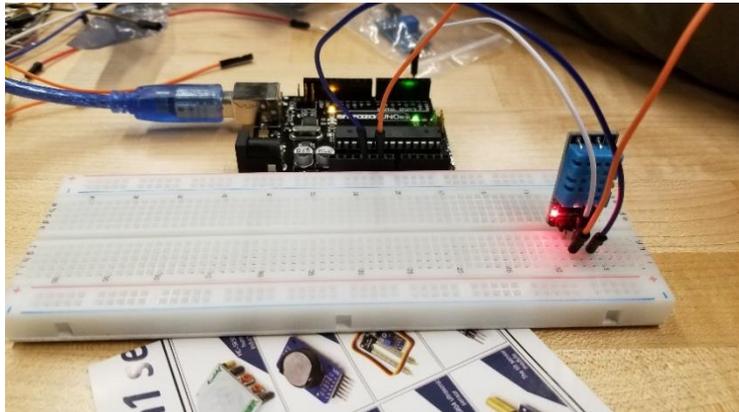
  // DHT11 sampling rate is 1Hz.
  delay(2500);
}

Serial uploading
```

b) Serial monitor:



c) Actual look of testing:



- C. What information is being measured:
The correctness of the codes for each type of sensors, of the circuit connection between Arduino board and sensors, of functionality of sensors.
- D. What is being observed and how is it being recorded:
The codes and circuit connections for all three types of sensors are correct, and the sensors are functional and working as we expected. The process are recorded by pictures and screenshots as provided above in testing process, and the screenshots of serial monitor are the proofs for the correct functionalities of our sensors.

E. Required materials and costs:

Material	Cost (CAD)
Detective sensors	\$ 12.54
Temperature sensor & Infrared sensor	\$ 39.99
Arduino board & other accessories	\$ 38.99
Discounts	\$ 9.25
Total	\$82.27

F. What work needs to be done:

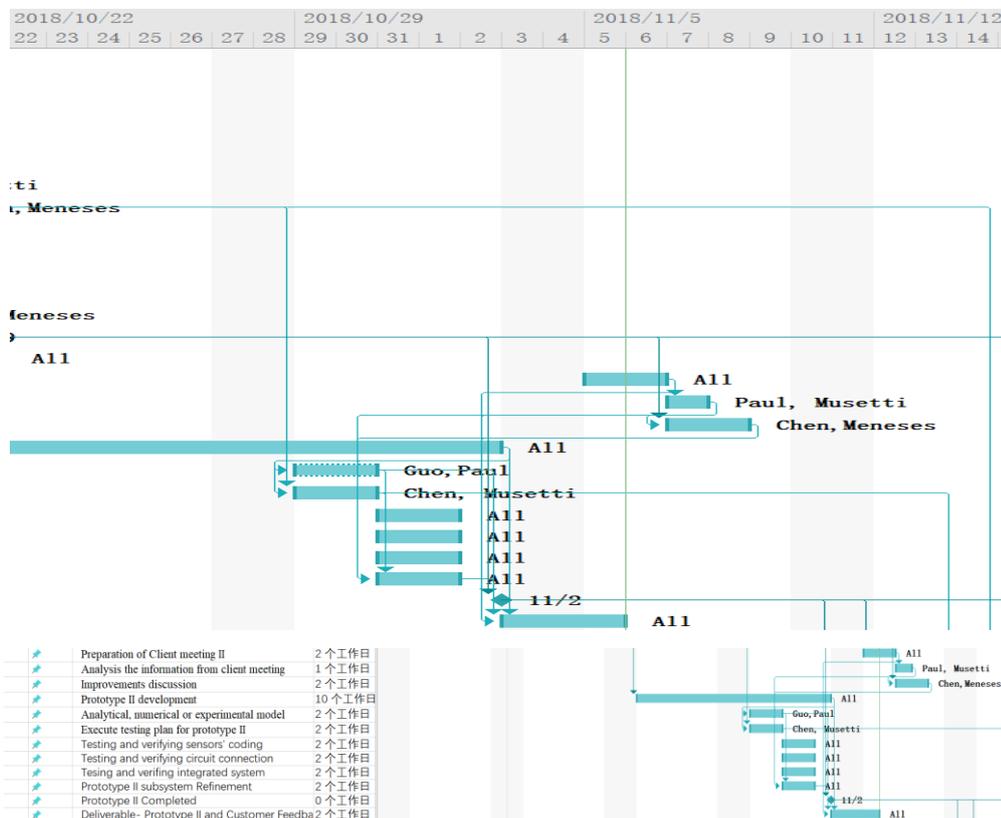
- 1) Order and get the materials as above described.
- 2) Research for relative coding, circuit connections, and download the Arduino software for programming.

4) When is it happening:

A. How long will the test take and what are the dependencies?

This period of prototype II development is 10 days, which includes the coding for sensors that needs to be done before testing. The testing period is 2 days.

B. A separate test planning Gantt chart:



C. When are the results required:

The results are required on November 3, since this deliverable is depending on the testing process and results, as well as we need to ensure we reach the milestone of prototype II completed to start working on our third prototype.

ANALYSIS OF CRITICAL COMPONENTS OR SYSTEMS:

Material being used: Arduino board

Why use Arduino board?

- Inexpensive compared to other microcontroller platforms
- Most microcontroller systems run on only windows. Arduino on the other hand runs on windows, Macintosh OSX, and Linux operating systems
- Simple and easy to use for beginners, yet flexible enough for more advanced users

Properties of Arduino UNO Board:

Properties	Values
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14
Analog Input pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
LED Built-in	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Risks and problems

The coding of sensors and circuit connections between sensors and Arduino can present several risks or problems. One of these risks is that, while making the robot it is common for shavings to get into the board. Anything left lying around in the same compartment as the Arduino can end up shorting something. It should be kept well covered in order to prevent this from happening. Another risk is one that can easily occur which is changing connections while the Arduino is powered on. This should never be done because it makes it very easy to short something. Furthermore, overloading a pin can be a risk. The pins current limit should be known or else there

is the danger of burning out the pin or the entire board. The total current being used by the sensors attached to the Arduino board should be found as well as the current limit for the Arduino board. If the total current is more than the boards current limit, then there is the risk of frying the board. To solve this problem an external power source may be required. Knowing how much power your components need compared to how much the board can provide is essential. Another risk is connecting more than 6 volts to any of the pins because it can overload it. Also, Arduino boards tend to heat up, affecting the temperature of the air surrounding it. This can affect the functionality of some of the sensors. Moreover, a risk is that the Arduino board should not be rested on conductive surface. It could cause it to short out. Another risk or problem is that wires can easily be mixed up. In order to prevent this from happening, a consistent wiring color code should be kept. For example, using red wires for power and black for ground. Additionally, another risk or problem would be connecting LEDs directly to addition pins without a current limiting resistor. Finally, a problem could be syntax errors in the code. There are many mistakes to be made when coding and the most typical errors include: missing semicolons at the end of lines, missing/wrong type of brackets, spelling errors, etc. Any of these small errors can stop the code from working.

CUSTOMER FEEDBACKS:

Our third client meeting will take place in a couple days and our second prototype will have been completed and tested by that time. After our first two client meetings, we were able to gather sufficient information on Bowie that lead to the idea of a robot braking system using sensors. The client meetings allow us to get a clear view of the pros and cons of the robot and our current ideas, and we gain further knowledge in terms of the process and path we should take throughout the creation of our prototype.

Since this will be the first time that our client will see the actual hands-on prototype, we will be able to get valuable information to improve our prototype. We will do this by empathizing with our client, since she is aware of all the product needs. We will gather as much feedback as we can, and implement the crucial (very necessary) bits to our prototype and solution.

LIFE-LONG LEARNING: TRANSFER OF KNOWLEDGE:

In this deliverable, we apply the same knowledge as last deliverable: the differences between different types of prototype, how to choose the most suitable prototype that will meet our expectation, how-to device a testing plan and execute it to test the fidelity of our prototype, and how to receive and apply feedbacks. Furthermore, we choose comprehensive and analytic prototype is due to the reason that we are working on an intergraded system for the coding of different types of sensors, and how will them work together by Arduino.

CONCLUSION:

We develop a comprehensive analytical prototype to verify the functionality of our sensors, especially the coding for integrated system and the circuit connections of Arduino. After executed our testing plan, our team concludes that the coding for each individual type of sensor is correct, but there are some improvements for the integrated coding which three types of sensors working together. Furthermore, the circuit connections of Arduino are correct and have been proved by our test results. In addition, the exact values of detective distances and detective period will be verified during next client meeting and applied on next prototype. Finally, we plan to present our prototype II in next client meeting .