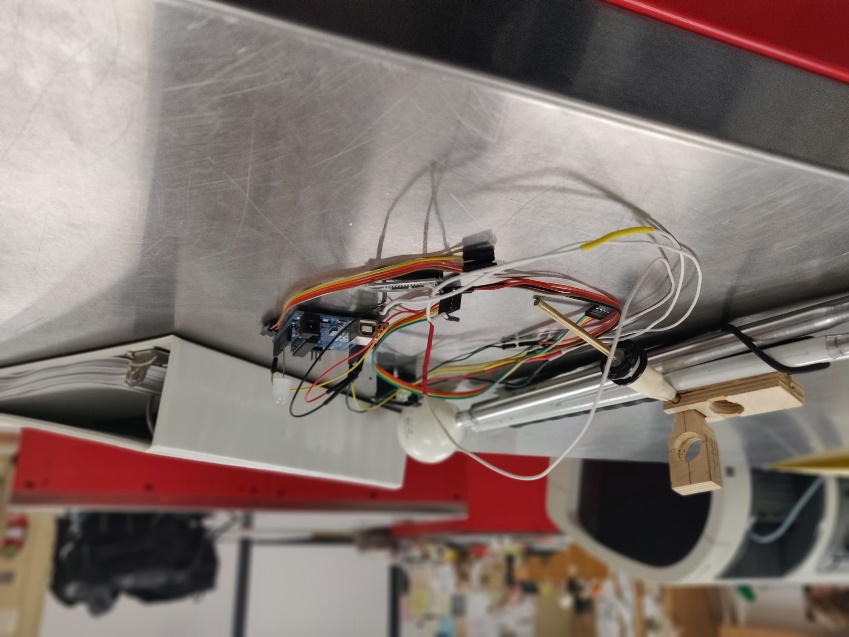
**Niskayuna Senior Research and Development**

**VisiStick**

Engineering Notebook

2018-2019

****

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| Subject |  | Page # |
| Cover/Title Page |  | 1 |
| Table of Contents |  | 2 |
|  |  |  |
| **Part One - The Team** |  | 3 |
| Meet the Team |  | 4 |
| Challenges |  | 5 |
|  |  |  |
| **Part Three - Engineering Sketchbook** |  | 6 |
|  |  |  |
| **Part Four – Programs and Algorithms** |  | 8 |
|  |  |  |
| **Part Five – Datasheets and Resources** |  | 10\* |
|  |  |  |

**Part One**

The Team

In this section, we will:

* Introduce the members of our team
* Describe the community and values which we represent
* Give the goals of what we hope to accomplish
* Explain our involvement in the world around us

**Meet the Team**

|  |  |
| --- | --- |
| **Seth Teichman**  Programming, Electrical, Documentation  I am a senior at Niskayuna High School. I took the class because of my fascination with computer science and engineering. I am currently our systems administrator and lead programmer. Outside of R&D, I also work as a member of the Robotics, IT Help Desk, and Recycling Clubs. Additionally this year, I am also taking Advanced Manufacturing. | **Kyle Ayers**  Mechanical, Financial    Insert Bio Here |

**Challenges**

|  |  |
| --- | --- |
| Date | Summary |
| 11.02.2018 | Today, we got a response back from the person we are developing the VisiStick for, and we learned that she uses iOS. For Seth, our programmer, iOS development is going to be completely new for him, since he has only written for Android, Windows, and Linux in the past. |
| 11.06.2018 | After doing some research into iOS development, Seth found out that he will need a Mac to build the final app, and for most development functions as well, since the iOS SDK only runs on Mac OS X. While there is a single Mac in the Engineering Room, it does not have XCode installed, so either he will have to get an XCode installed to it by the IT/MTS Department, or we will have to get our hands on a non-district Mac device. |
| 12.03.2018 | In all of our past work, Seth has been using a HC-06 Bluetooth module, yet upon further research, it is only compatible with Windows and Android because Serial over Bluetooth is not supported on iOS, yet the person we are developing for uses iOS. We would like to be able to expand our support to Android after finishing development specifically for her, so he needs to find a Bluetooth LE module that will run on 3.3V Line Level. |
| 12.13.2018 | For the past week, Kyle has been working on making 3D Models for cases for various parts of our current design. His current model of a case for the Lidar sensor is a solid object, we would have to insert the sensor during the printing process, and in the event that the case needed to be redesigned in the future, we would have had to break the part to get the sensor out. He is now redesigning it such that it latches together, so we do not run into any problems during development. |
| 12.21.18 | After proposing his idea for the “Bearing and Mount” design to Seth, Kyle found that the design would not constantly be facing forward, something that Seth required to get values consistent with the actual position of the module. |
| 01.07.19 | Now that we have received the new Bluetooth LE module, as well as other components, Seth needs to rewrite his previous code to work with the new Bluetooth LE module, which uses UART instead of Serial. This does make development easier overall, as it is much more supported, and is supported on iOS, whereas Serial is not. |

**Part Three**

Engineering Sketchbook

In this section, we will:

* Provide updates on specific designs we’ve come up with, including sketches, and photos, if they were brought to fruition

General Design

|  |  |  |
| --- | --- | --- |
| Design | Pros | Cons |
| Ball Bearing and Pivot (11/17/18-12/18/18) | * Compact * Keep the distance Sensor facing the desired Direction | * Heavier * Might be too prone to move when the entire cane is moved |
| Bearing and Mount (12/19/18-12/21/18) | * Lighter * More compact * Sensor moves independent of the bearing (A) * Counteracts more movement than previous design | * Not consistently facing forward * Would have to add another mechanism to get it to face forward |
| Sensor Case (12/19/18-) | * Lightweight * Protection on all sides * Easily replaceable | * May collect condensation |

|  |  |  |
| --- | --- | --- |
| Bearing and U-Mount (12/21/18-2/11) | * Faces in the right direction | * Heavier than previous design * Weight has to be at least two times as much as that of the Arduino and power modules * Must have loose wires for sensor |
| Bearing and U-Mount v2 (2/11-2/26) | * Faces in the right direction * Removes tension from wires, allowing free movement from gravity | * More complex than Bearing and U-Mount * Internal Conductive Lines must be separate in a small amount of space * The tape used for insulation melted too fast to solder and smooth the copper tape used in the conductive lines |
| Bearing and U-Mount v3.0P (3/8/19-3/18/19) | * Faces in the right direction * Removes tension from wires, allowing free movement from gravity * Easily assembled, no permanent fastener * Higher quality product | * Requires more machined parts * Weird dimensions because T-Spacers are designed around the randomly chosen dimensions for the rest of the case, making it hard to produce these reliably |
| Bearing and U-Mount v3.5 (3/18/19-) | * Pros of v3.0 * Redesigned and redimensioned around the more regular dimensions chosen for the reproducibility of the T-Spacers * All T-Spacers are the same, there are no unique spacers * Spacers inserted into bearing instead, because it is a press fit |  |

**Part Four**

Programs and Algorithms

In this section, we will:

* Provide a full copy of all of our code, both from ideating and prototyping
* A detailed description of any unique algorithms we have written or implemented

Ideating

*Arduino with Lidar (Arduino\_with\_Single\_Lidar.ino) – 10/23/18*

void setup()

{

Serial.begin(9600);

pinMode(13, OUTPUT);

//RED -> 5V

//BLACK -> GND

//YELLOW -> A0

}

void loop()

{

digitalWrite(13, HIGH);

int val = analogRead(A0); // read the sensor (inches)

Serial.println(exp(8.5841 - log(val)));

delay(50);

}

*Arduino with Bluetooth (Arduino\_with\_Bluetooth.ino) – 10/25/18*

//RXD -> TX/1

//TXD -> RX/0

//GND -> GND

//VCC -> 3.3V

char bluetoothVal; //value sent over via bluetooth

char lastVal; //stores last state of device (on/off)

char mode='n'; //d=Data, c=Command

int count = 0;

int readings[3];

void setup()

{

mode='n';

Serial.begin(9600);

pinMode(13,OUTPUT);

}

void loop()

{

if(Serial.available()) {

bluetoothVal=Serial.read();

}

if (mode=='n') {

Serial.println(F("Select Mode [(C)ommand or (D)ebug]:"));

if (bluetoothVal=='c') {

Serial.println(F("Command Mode Selected"));

mode='c';

lastVal=bluetoothVal;

} else if (bluetoothVal=='d') {

Serial.println(F("Debug Mode Selected"));

mode='d';

lastVal=bluetoothVal;

} else {

if (bluetoothVal!=lastVal) {

Serial.println(F("Invalid Option. [(C)ommand or (D)ebug]:"));

}

}

delay(2500);

} else if (mode=='c') {

if (bluetoothVal=='b') {

mode='n';

Serial.println(F("Exited Command Mode"));

lastVal=bluetoothVal;

}

if (bluetoothVal=='0' && lastVal!=bluetoothVal) {

digitalWrite(13, LOW);

Serial.println(F("LED toggled Off"));

lastVal=bluetoothVal;

} else if (bluetoothVal=='1' && lastVal!=bluetoothVal) {

digitalWrite(13, HIGH);

Serial.println(F("LED toggled On"));

lastVal=bluetoothVal;

}

} else if (mode=='d') {

if (bluetoothVal=='b') {

mode='n';

Serial.println(F("Exited Debug Mode"));

lastVal=bluetoothVal;

}

count++;

Serial.println(count);

delay(50);

}

}

*Arduino with Lidar and Bluetooth (Arduino\_with\_Single\_Lidar\_and\_Bluetooth\_v1.ino) – 10/29/18*

//RXD -> TX/1

//TXD -> RX/0

//GND -> GND

//VCC -> 3.3V

char bluetoothVal; //value sent over via bluetooth

char lastVal; //stores last state of device (on/off)

char mode='n'; //d=Data, c=Command

int count = 0;

int readings[3];

void setup()

{

mode='n';

Serial.begin(9600);

pinMode(13,OUTPUT);

}

void loop()

{

if(Serial.available()) {

bluetoothVal=Serial.read();

}

if (mode=='n') {

Serial.println(F("Select Mode [(C)ommand or (D)ebug]:"));

if (bluetoothVal=='c') {

Serial.println(F("Command Mode Selected"));

mode='c';

lastVal=bluetoothVal;

} else if (bluetoothVal=='d') {

Serial.println(F("Debug Mode Selected"));

mode='d';

lastVal=bluetoothVal;

} else {

if (bluetoothVal!=lastVal) {

Serial.println(F("Invalid Option. [(C)ommand or (D)ebug]:"));

}

}

delay(2500);

} else if (mode=='c') {

if (bluetoothVal=='b') {

mode='n';

Serial.println(F("Exited Command Mode"));

lastVal=bluetoothVal;

}

if (bluetoothVal=='0' && lastVal!=bluetoothVal) {

digitalWrite(13, LOW);

Serial.println(F("LED toggled Off"));

lastVal=bluetoothVal;

} else if (bluetoothVal=='1' && lastVal!=bluetoothVal) {

digitalWrite(13, HIGH);

Serial.println(F("LED toggled On"));

lastVal=bluetoothVal;

}

} else if (mode=='d') {

if (bluetoothVal=='b') {

mode='n';

Serial.println(F("Exited Debug Mode"));

lastVal=bluetoothVal;

}

int val = analogRead(A0); // read the sensor (inches)

readings[count] = val;

if (exp(8.5841 - log(val)) > 10 && exp(8.5841 - log(val)) < 85) {

count++;

}

if (count == 2) {

int math[3] = {exp(8.5841 - log(readings[0])), exp(8.5841 - log(readings[1])), exp(8.5841 - log(readings[2]))};

int fval = (math[0] + math[1] + math[2])/3;

Serial.println(fval);

count = 0;

}

delay(50);

}

}

**Part Five**

Datasheets and Resources

In this section, we will:

* Include most parts of the datasheets for the various electronic parts that we will be using to bring the VisiStick into existence