My GPS Tracker

By: Isabella Jean Morison

Hypothesis and Testable Question

I am very scattered, and so I lose things a lot. This project is aiming at helping me find those things using GPS.

If I make this GPS tracking sticker and attach it to important items, then I will be able to find them easier with the help of GPS and associated app, because I can triangulate the exact location of my lost objects with GPS.

Materials

- 1 GPS Receiver
- 1 Uno R3 Controller Board
- 1 830 Tie Points Breadboard
- 1 Potentiometer
- 1 LCD1602 Module
- 20 Breadboard Jumper Wires
- 1 USB Cable
- 1 GPS Controller
- Arduino Program installed on computer
- A computer

Procedure

- First I bought and assembled all necessary materials.
- Next, I put together the component that would detect my current gps location.
- Next, I assembled the code to display my location to my computer.
- After this, I wired and built the LCD display, which would later display my GPS location.
- Then I coded the LCD display to display my location.

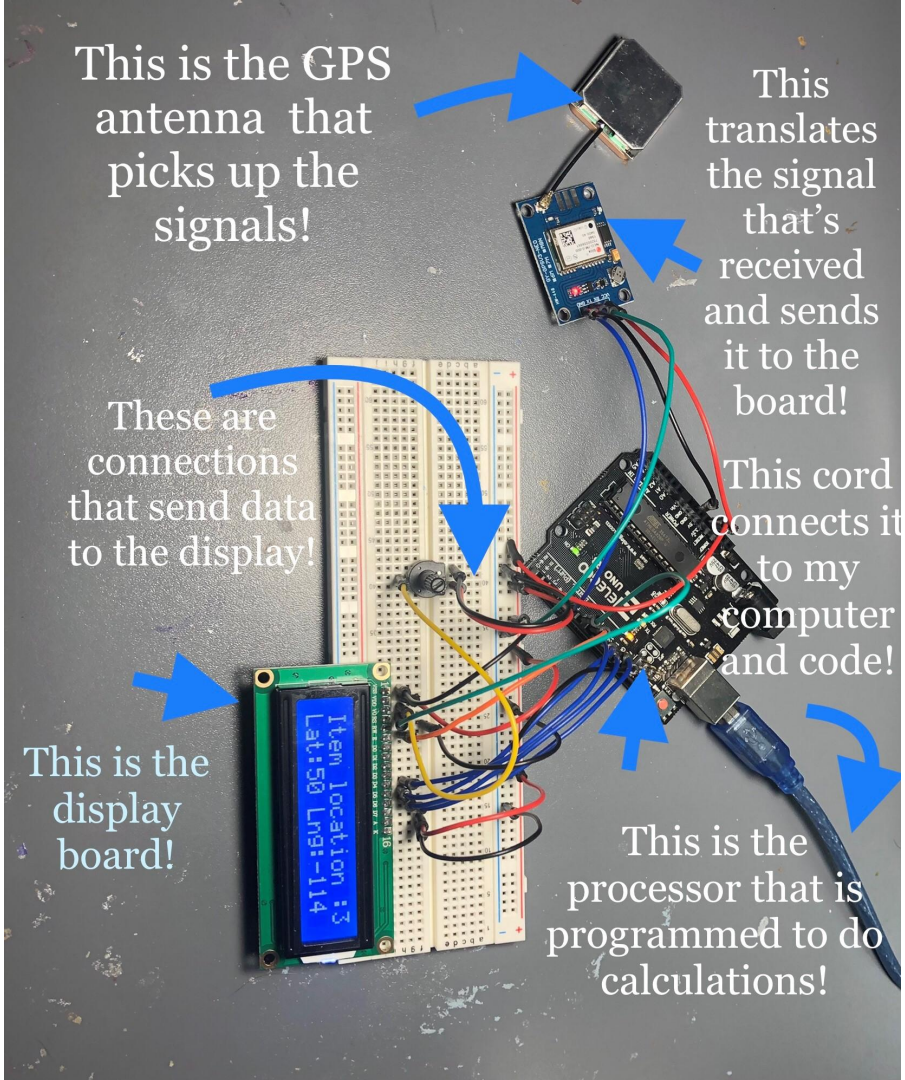

Diagram of Build

Here is my diagram of my GPS model!

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
/*
   This sketch uses code from a sample TINYGPS++ and LCD sketches
   It requires TINYGPS++.h, SoftwareSerial.h and LiquidCrystal.h
*/
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
static const int RXPin = 4, TXPin = 3;
static const uint32_t GPSBaud = 9600;
int LAT;
int LNG;
// The TinyGPS++ object
TinyGPSPlus gps;
// The serial connection to the GPS device
SoftwareSerial ss(RXPin, TXPin);
void setup()
{
    // initialize serial
  Serial.begin(9600);
  ss.begin(GPSBaud);
    // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Item location :3");
}
void loop()
{
```

```
void loop()
{
  // This sketch displays information every time a new sentence is correctly encoded.
  while (ss.available() > 0)
    if (gps.encode(ss.read()))
      displayInfo();
  if (millis() > 5000 && gps.charsProcessed() < 10)
  {
    Serial.println(F("No GPS detected: check wiring."));
    while(true);
  }
}
void displayInfo()
{
  Serial.print(F("Location: "));
  if (gps.location.isValid())
  {
    Serial.print(gps.location.lat(), 6);
    Serial.print(F(","));
    Serial.print(gps.location.lng(), 6);
    LAT = gps.location.lat();
    LNG = gps.location.lng();
    lcd.setCursor(0, 1);
    lcd.print("Lat:");
    lcd.print(LAT);
    lcd.print(" ");
    lcd.print("Lng:");
    lcd.print(LNG);
  }
```

```
    lcd.print(LNG);
  }
  else
  {
    Serial.print(F("INVALID"));
  }
  Serial.print(F("  Date/Time: "));
  if (gps.date.isValid())
  {
    Serial.print(gps.date.month());
    Serial.print(F("/"));
    Serial.print(gps.date.day());
    Serial.print(F("/"));
    Serial.print(gps.date.year());
  }
  else
  {
    Serial.print(F("INVALID"));
  }
  Serial.print(F(" "));
  if (gps.time.isValid())
  {
    if (gps.time.hour() < 10) Serial.print(F("0"));
    Serial.print(gps.time.hour());
    Serial.print(F(":"));
    if (gps.time.minute() < 10) Serial.print(F("0"));
    Serial.print(gps.time.minute());
    Serial.print(F(":"));
    if (gps.time.second() < 10) Serial.print(F("0"));
```

```
  if (gps.time.hour() < 10) Serial.print(F("0"));
  Serial.print(gps.time.hour());
  Serial.print(F(":"));
  if (gps.time.minute() < 10) Serial.print(F("0"));
  Serial.print(gps.time.minute());
  Serial.print(F(":"));
  if (gps.time.second() < 10) Serial.print(F("0"));
  Serial.print(gps.time.second());
  Serial.print(F("."));
  if (gps.time.centisecond() < 10) Serial.print(F("0"));
  Serial.print(gps.time.centisecond());
}
else
{
  Serial.print(F("INVALID"));
}
Serial.println();
```

```
2:33:23.020 -> Location: 50.927032,-114.027000  Date/Time: 1/7/2021 19:33:22.00
2:33:23.114 -> Location: 50.927032,-114.027000  Date/Time: 1/7/2021 19:33:22.00
2:33:23.400 -> Location: 50.927032,-114.027000  Date/Time: 1/7/2021 19:33:22.00
2:33:23.495 -> Location: 50.927032,-114.027000  Date/Time: 1/7/2021 19:33:22.00
2:33:24.014 -> Location: 50.927032,-114.027000  Date/Time: 1/7/2021 19:33:23.00
2:33:24.155 -> Location: 50.927032,-114.027000  Date/Time: 1/7/2021 19:33:23.00
2:33:24.389 -> Location: 50.927032,-114.027000  Date/Time: 1/7/2021 19:33:23.00
2:33:24.528 -> Location: 50.927032,-114.027000  Date/Time: 1/7/2021 19:33:23.00
2:33:24.996 -> Location: 50.927032,-114.027000  Date/Time: 1/7/2021 19:33:24.00
2:33:25.138 -> Location: 50.927032,-114.027000  Date/Time: 1/7/2021 19:33:24.00
2:33:25.372 -> Location: 50.927032,-114.027000  Date/Time: 1/7/2021 19:33:24.00
2:33:25.513 -> Location: 50.927032,-114.027000  Date/Time: 1/7/2021 19:33:24.00
2:33:25.984 -> Location: 50.927032,-114.027000  Date/Time: 1/7/2021 19:33:25.00
2:33:26.125 -> Location: 50.927032,-114.027000
```

Diagram of Code

This is my code!
It has comments that explain the things within the code, but it gives directions to my model!

Observations

First, I took the GPS receiver and attached the end of its antenna to the left side of the GPS controller. It snapped together. Next, I took four jumper wires. On the right side of the GPS controller there was 4 holes-one labeled VCC (for power or voltage common collector), one labeled RX (for receive), one labeled TX (for transmit), and the last labeled GND for ground. I took one of the wires and soldered it in the VCC hole, and solder the last three to RX, TX, and GND. Then take the Uno R3 board, and take the wire that is connected the VCC and inserted it in one of the holes on the left side of the Uno board that is labeled 3.3V. I connected the RX wire and inserted it into the right side of the Uno board, in the hole labeled -3. I took the TX wire and connect it to the 4 on the right side of the Uno board, beside the RX wire. I connected the GND wire and connect it to the hole on the left side of the Uno board labeled GND. Then I went to: Get GPS location from U-Blox Neo-6 and Neo-7 GPS Modules with Arduino - Robojax and read through the code there and the comments, which shows how it works. Then I went to: TinyGPS++ | Arduiniana. I scrolled down until you're at the download and installation section. I followed the instructions for downloading TinyGPS++, which is needed in the code. Then I went into arduino. I deleted the code that was already there. Then, I pasted the code from before. I pressed the symbol in the top right corner, the serial monitor. A new screen popped up. It started by showing constant messages of numbers, followed by "Location: INVALID," and the Time/Date. After about a minute or so, it started showing my location.

Observation Part 2

Then I went to: ELEGOO UNO Project Super Starter Kit Tutorial - ELEGOO Official, and downloaded the PDF. I went to lesson 14, LCD Display. I grabbed the LCD model, the 16 other wires, and the breadboard. I looked at the diagrams and plugged in the wires as shown. Then, I went here: Arduino - LiquidCrystal and downloaded liquid crystal. I went into my code from before on Arduino. At the start of my code, I put #include <LiquidCrystal.h>. Then, right before the rest of my code after all of the #include statements, I put LiquidCrystal lcd(7, 8, 9, 10, 11, 12);. This initializes the library with the number of the interface pins. In void setup (), after the initialize serial code, I put lcd.begin(16, 2);, which sets up the LCD's number of columns and rows. Then after that I put: lcd.print("Item location"). This displays Item Location to the LCD. Then, under Void displayInfo (), after Serial.print(gps.location.lng(), 6);, I put LAT = gps.location.lat() LNG = gps.location.lng(); lcd.setCursor(0, 1); lcd.print("Lat:");lcd.print(LAT); lcd.print(" "); lcd.print("Lng:"); lcd.print(LNG);. Then, I clicked serial monitor, and after a minute or so, I saw my rounded location displayed on the LCD!

Analysis

My problem did not quite solve my problem. I missed one crucial part of my design, though- the wireless aspect that would be the last crucial step to make this project functional. So i didn't really build a tracking device that would solve my main problem and function as desired, but that is ok-because a logical next step I would take would be to build the wireless aspect of my device. To cross the gap in planning I unknowingly made at the start.

Sources of Error

My experiment did and did not answer my testable question, which was if I could find my lost things with GPS, and so that becomes a source of error. I did build a GPS model which records your current location, and a LCD board to display it; but it is all connected through wires. If I had a wireless model, then my project would bridge the gap of the separate component-the things you attach to your items and the model that displays data-but I couldn't do that all in one year, and so I plan to continue on with this project next year, and find a way to make it wireless. Also, I couldn't get the technology with precise precision-it used to be limited to U.S army's, and though it is available now, it is far too pricey. This project would still help finding things that are far away, though.

Conclusion

My innovation turned out to be inconclusive. I cannot find my objects with this device yet and solve my leading problem, because since the two parts of the device (the part that displays where your items are and the part that you connect to your items) are connected. Once I find a wireless way to do this, though, this should function as planned, and solve the leading problem, and so I am doing a science fair next year on this. The finding device still does not really have a way of adhesion yet, and so is not quite a sticker like the hypothesis said it should be. I *can* display the data of the current position of my device on an LCD board. It is, in a way, still a work in progress.

Extra Citations

Hart, Mikal, (N.D), *Tiny GPS++,* Arduiniana. Retrieved from: http://arduiniana.org/libraries/tinygpsplus/

Hart, Mikal, (Sep 9th, 2018), *Get GPS location from U-Blox Neo-6 and Neo-7 GPS Modules with Arduino,* Robojax. Retrieved from: https://robojax.com/learn/arduino/?vid=robojax_GPS_TinyGPSPlus

N.A, (Oct 21, 2020), *ELEGOO UNO PROJECT SUPER STARTER KIT TUTORIAL,* Elegoo. Retrieved from: https://www.elegoo.com/blogs/arduino-projects/elegoo-uno-project-super-starter-kit-tutorial

N.A, (N.D), *LiquidCrystal Library,* Arduino. Retrieved from: https://www.arduino.cc/en/Reference/LiquidCrystal