Arduino GPS Module

Overview

The sketches used in this tutorial utilizes the Adafruit Ultimate GPS Breakout (https://www.sparkfun.com/products/15569) module with an Arduino Uno to allow the user to discover their latitude and longitude at their position and to get directions to a set location from their current location. These tutorials are what could be considered the backbone of modern GPS systems used in cars, phones, and watches to guide people along their daily commutes.

For the rover mission in development, the use of the GPS module will guide the rover to position itself in the north direction in Mission 3 and to drop its payload at a specified longitude and latitude in the final mission.

Requirements

Hardware requirements to complete the sketches in this tutorial include: 1 Breadboard, 1 Arduino Uno, 1 USB cable A to B, 1 Adafruit Ultimate GPS Breakout with antenna, and 4 jumper wires (M/M).

Software requirements include the Arduino IDE program (or equivalent), and the TinyGPS++ Arduino library (https://github.com/mikalhart/TinyGPSPlus) installed in Arduino IDE.

Setup

1) First is to wire the hardware up as seen in Figure 1.
2) Connect the Arduino Uno to computer with USB Cable.
3) Open Arduino IDE and install the TinyGPS++ library.
4) Then upload the sketches into Arduino IDE and upload them to the Arduino to run the programs.

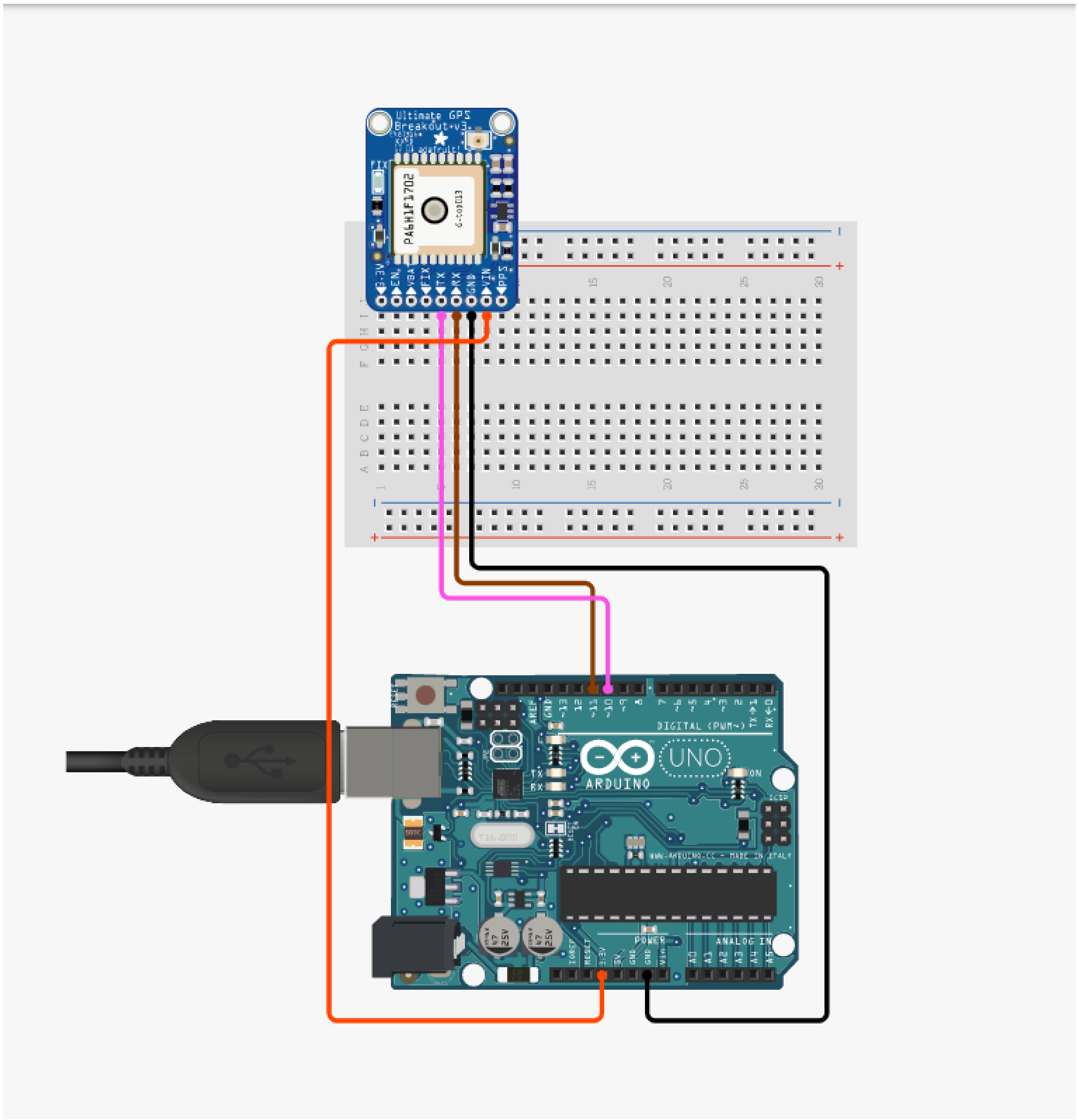

Figure 1:: Wiring diagram for the GPS module to Arduino.

Sketch 1: GPS-Lat and Long

This sketch will give the user’s current latitude and longitude coordinates through the serial monitor after a prompt from the serial monitor.

Code

```
//This arduino sketch utilizes the Adafruit Ultimate GPS Breakout - 66 channel w/10 Hz updates - PA1616S
```

```
//This sketch will show the current latitude and longitude of the user after the input from the serial port
//Reference https://makezine.com/projects/gps/ by  Mikal Hart



#include <SoftwareSerial.h>
#include <TinyGPS++.h> //Requires library for the GPS from https://github.com/mikalhart/TinyGPSPlus/blob/master/src/TinyGPS%2B%2B.h
#define RXPin 10   //the recieve pin on the arduino will go to the 'TX' pin on the GPS module
#define TXPin 11   //the transmistion pin(Pin3) on the arduino 'RX' pin on the GPS module
#define GPSBaud 9600  //sets baud rate for the gps module to arduino
#define ConsoleBaud 9600 //sets baud rate for the console/serial port of the arduino

// The serial connection to the GPS device
SoftwareSerial ss(RXPin, TXPin);

// The TinyGPS++ object
TinyGPSPlus gps;

void setup()
{
  Serial.begin(ConsoleBaud);
  ss.begin(GPSBaud);
}

void loop()
{
   while (ss.available() > 0)  //if any data is recieved from the GPS
```

```
    gps.encode(ss.read());    //send them to the TinyGPS++ object


  if (gps.location.isUpdated() || gps.altitude.isUpdated()) //if the gps location has been updated, change the gps location

  {

    Serial.println("Input any key to find your current latitude and longitude from the GPS module"); //user input prompt

  while(Serial.available()){Serial.read();}

   while (!Serial.available()) { }

    Serial.print("Location: ");

    Serial.print("Latitude ");

    Serial.print(gps.location.lat(), 6);  //prints the latitude from the GPS

    Serial.print(",");

    Serial.print("Longitude ");

    Serial.println(gps.location.lng(), 6); //prints the longitude from the GPS

    }

}
```

Sketch 2: GPS-Direction

This sketch will give the user directions to the CMU Bookstore by giving them the cardinal directions from their current location and how many meters away they are from it. These directions are given in the serial monitor after a prompt from the user.

Code

```
//This Arduino sketch will report the current distance and compass heading to a specific destination (Currently the CMU Bookstore) at the users input on the serial monitor.

//The location in the sketch can be changed to any location with a change of lat/long and variables/outputs


#include <SoftwareSerial.h>

#include <TinyGPS++.h>    //Requires library for the GPS from https://github.com/mikalhart/TinyGPSPlus/blob/master/src/TinyGPS%2B%2B.h
```

```
#define RXPin 10          //the recieve pin on the arduino will go to the 'TX' pin on the GPS module
#define TXPin 11          //the transmistion pin(Pin3) on the arduino 'RX' pin on the GPS module
#define GPSBaud 9600      //sets baud rate for the gps module to arduino
#define ConsoleBaud 9600  //sets baud rate for the console/serial port of the arduino

SoftwareSerial ss(RXPin, TXPin);

// The TinyGPS++ object
TinyGPSPlus gps;
unsigned long lastUpdateTime = 0;

#define CMUBOOKSTORE_LAT 39.080819
#define CMUBOOKSTORE_LNG -108.552704



void setup()
{
  Serial.begin(ConsoleBaud);
  ss.begin(GPSBaud);
}

void loop()
{

  while (ss.available() > 0)     //if any data from the GPS has arrived save in the TinyGPS++ object
```

```
    gps.encode(ss.read());


  if (millis() - lastUpdateTime >= 5000) //wait and update every 5 seconds
  {
    lastUpdateTime = millis();


    //Comparison functions of current location and bookstore
    double distanceToDestination = TinyGPSPlus::distanceBetween(
      gps.location.lat(), gps.location.lng(),CMUBOOKSTORE_LAT,
CMUBOOKSTORE_LNG); //compares current lat/long with CMU Bookstore's to find
distance
    double courseToDestination = TinyGPSPlus::courseTo(
      gps.location.lat(), gps.location.lng(), CMUBOOKSTORE_LAT,
CMUBOOKSTORE_LNG); //compares current lat/long with CMU Bookstore's to find
heading
    const char *directionToDestination = TinyGPSPlus::cardinal(courseToDestination);
//takes difference from heading and transfers to cardinal directions



Serial.println("Input any key to find your current distance and required heading to the
CMU Bookstore"); //user input prompt
  while(Serial.available()){Serial.read();}
   while (!Serial.available()) { }



    if (distanceToDestination <= 10.0) //if less than 10 meters from bookstore report that
the user has arrived
    {
      Serial.println("You have arrived at the CMU Bookstore.");
      exit(1);
    }
```

```
     //Outputs the distance and heading to the bookstore
   Serial.print("You are ");
   Serial.print(distanceToDestination);
   Serial.println(" meters from the CMU Bookstore.");
   Serial.print("Head ");
   Serial.print(directionToDestination);
   Serial.println(" to go to the CMU Bookstore.");


  }
}
```

References

This tutorial was created with help from https://makezine.com/projects/gps/