

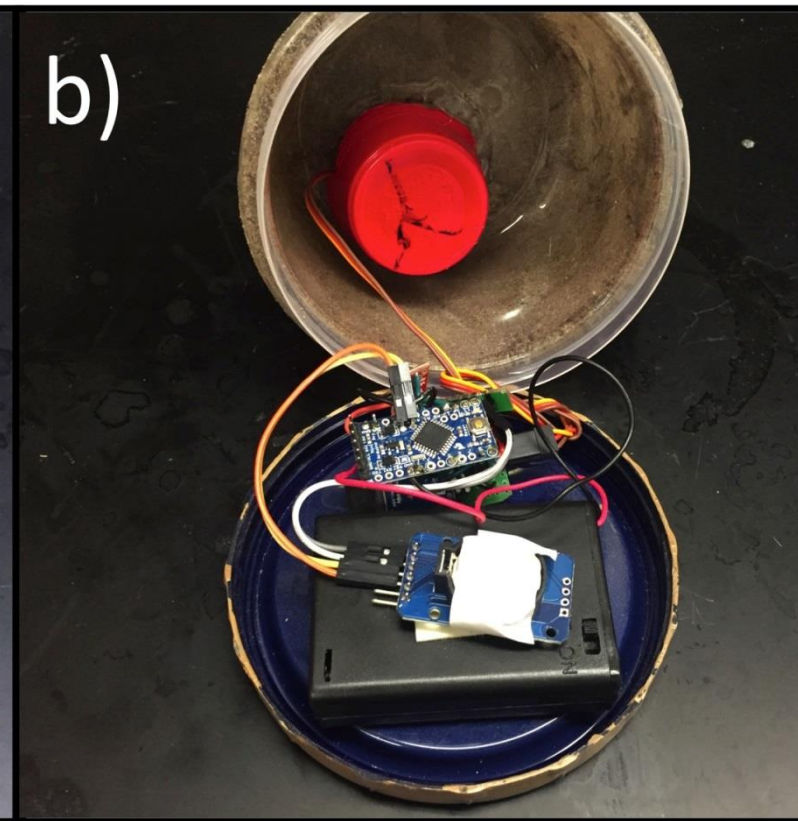
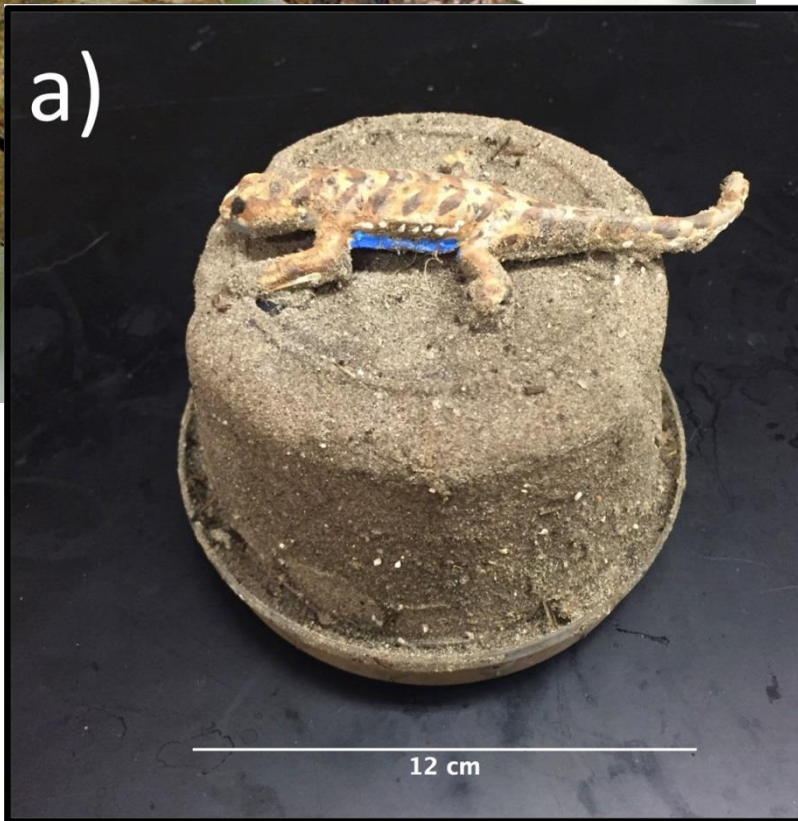
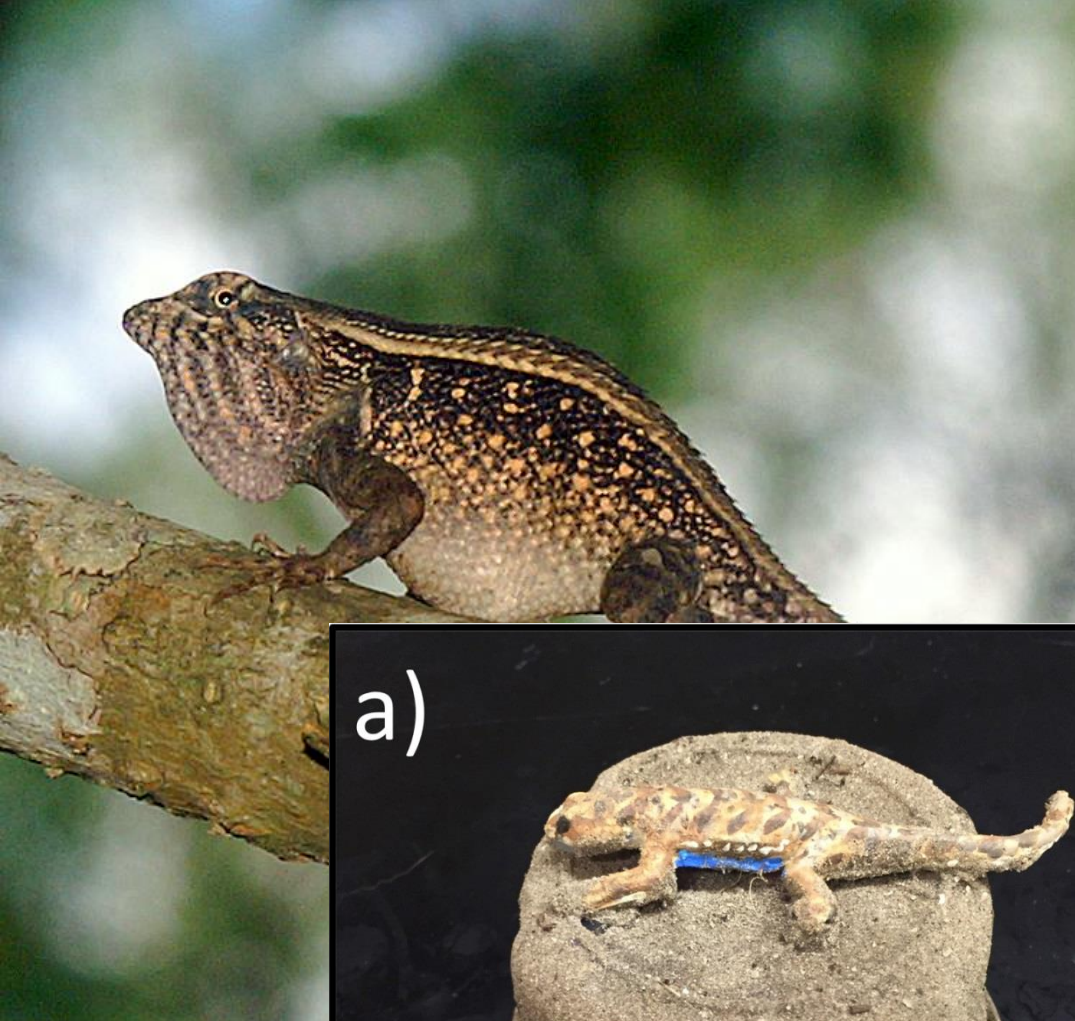
DIY GPS loggers for use in field studies



Pat Cain
Matt Cross



From the
beginning...

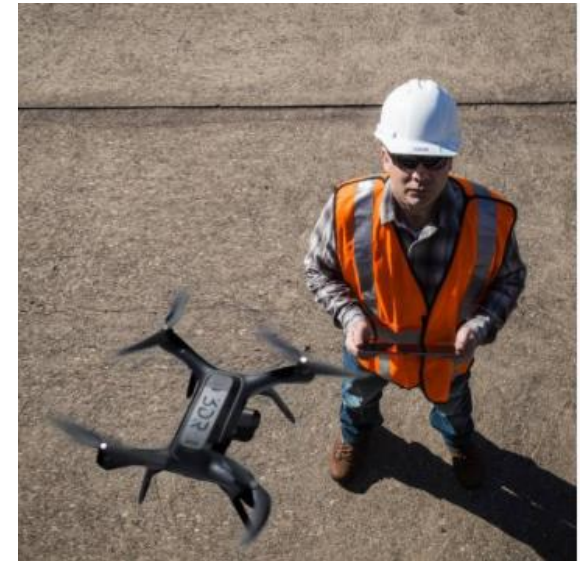


What is open source?

Linux



Mozilla



3D Robotics



microstill

[Home](#)

[About](#)

[Make it](#)

[Blog](#)

[Shop](#)

Make. Open Source. Moonshine

Build your own microstill and prepare spirits at home. All information just one click away.

[Learn more](#)

[Start Now](#)

➤ See blog

```
include <Bounce2.h>

pump relais
const int pump=9;

pump status, 0 for off, >0 for on
pumpstate = 0;
pumplevel 3,2,1,0 corresponding to 1000 ms, 1250 ms, 1700 ms, 2500 ms
pumplevel = 3;
pumppause of 1000, 1250, 1700, 2500 ms corresponding to pumplevel
signed long pumppause = 1000;

heater relais
const int heater=10;
```

Arduino Code ver0.2

In the last months significant improvements were made on the code. Today ver 0.2 is released. The main new features...

➤ Read more



Drawings Issue 02

Two months ago the drawings issue 01 was posted. Making the microstill I found typos and errors that needed to...

➤ Read more



Birdwatchers – Recipe of the Month

Birdwatchers is a simple recipe for sugar wash. Sometimes it is referred to as TPR (tomato past recipe). It produces a...

➤ Read more

What is Arduino?

“Arduino is an open-sourced electronics prototyping platform based on flexible, easy-to-use hardware and software. Its intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments”.

<http://arduino.cc>



What is Arduino?

- Atmega chip: microcontroller
- Breakouts: easy access to microcontroller
“microcontroller kits”
- Arduino open-source hardware
- Arduino programming language



```
Blink | Arduino 1.0
File Edit Sketch Tools Help

Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 * This example code is in the public domain.
 */




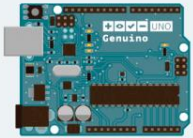


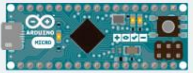
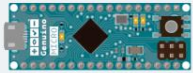



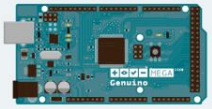


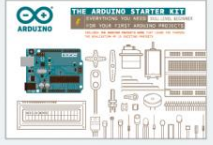

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

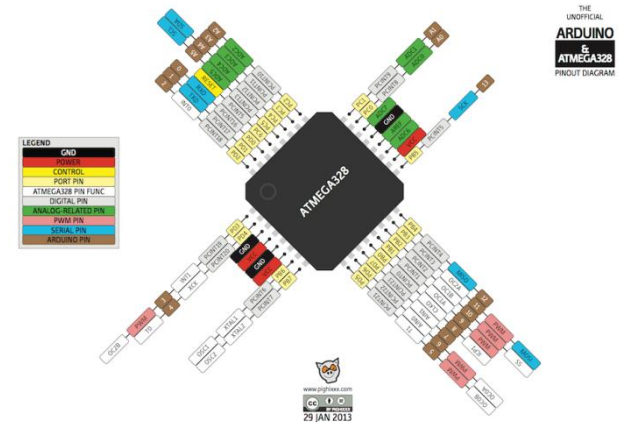
void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);            // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}

Arduino Uno on IDEv1.0.1
```


Arduino is two parts

1. Hardware

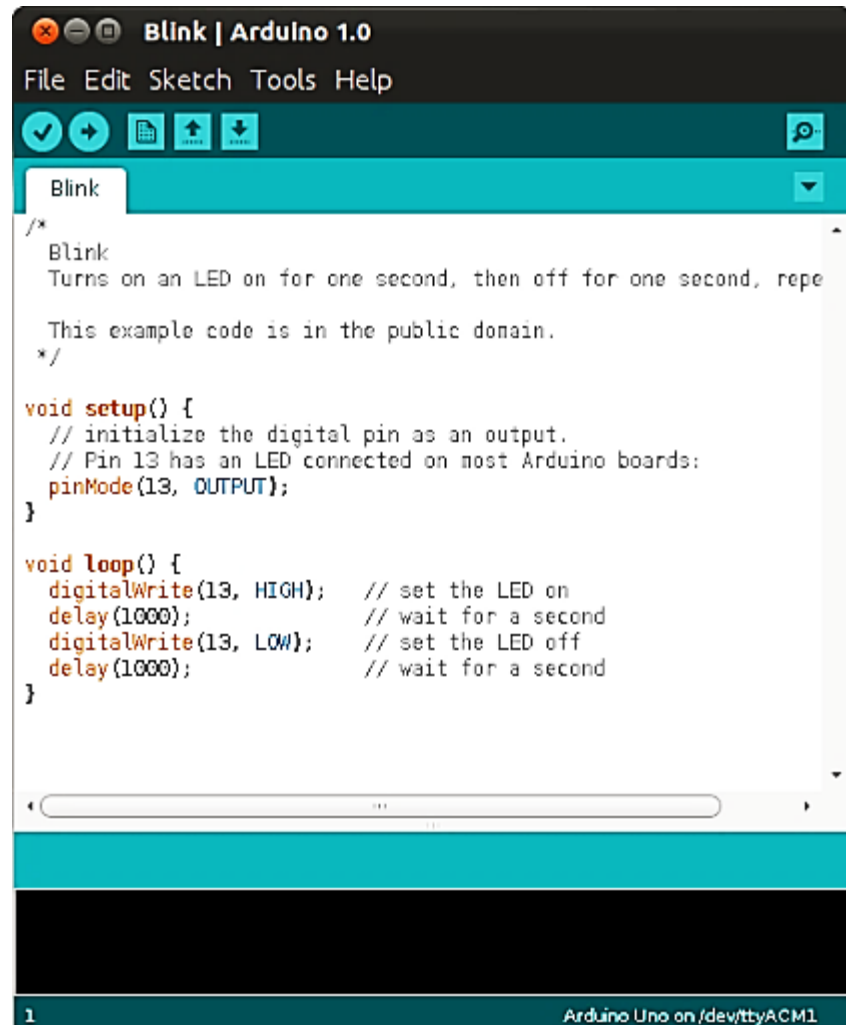
UNITED STATES	REST OF THE WORLD
 ARDUINO UNO	 Genuino UNO
	
 ARDUINO MICRO	 Genuino MICRO
	
 ARDUINO MEGA 2560	 Genuino MEGA 2560
	
 ARDUINO STARTER KIT	 Genuino STARTER KIT
	



Arduino is two parts”

1. Hardware

2. Software



The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for saving, running, uploading, and downloading. The sketch name "Blink" is displayed in the top left. The code editor contains the following C++ code:

```
/*  
 * Blink  
 * Turns on an LED on for one second, then off for one second, repeatedly.  
 *  
 * This example code is in the public domain.  
 */  
  
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);   // set the LED on  
  delay(1000);              // wait for a second  
  digitalWrite(13, LOW);   // set the LED off  
  delay(1000);              // wait for a second  
}
```

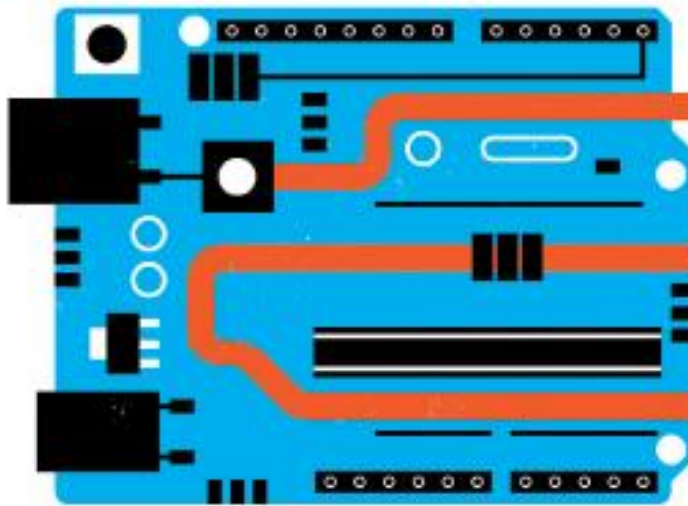
At the bottom of the window, the status bar shows "1" on the left and "Arduino Uno on /dev/ttyACM1" on the right.

TOOLBOX

AGE OF THE ARDUINO

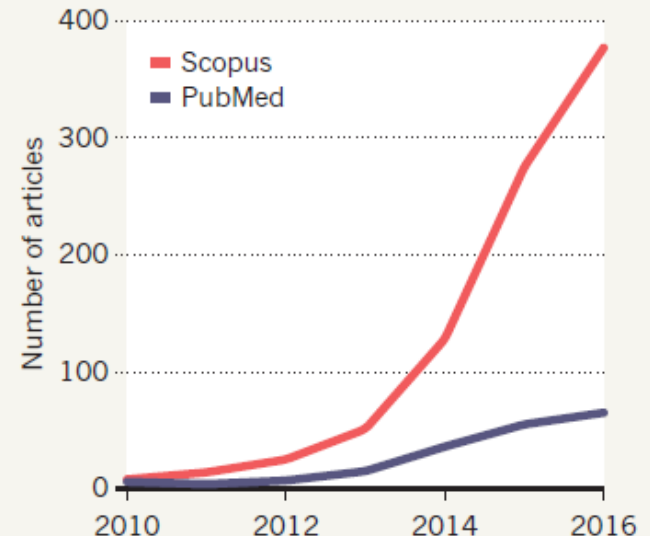
A booming market for small, cheap 'single-board computers' offers researchers a helping hand in fields such as automation, networking and data collection.

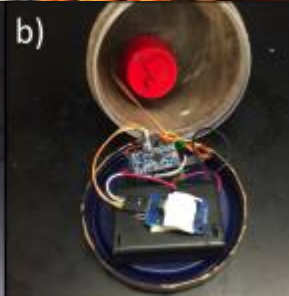
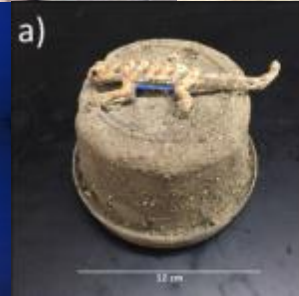
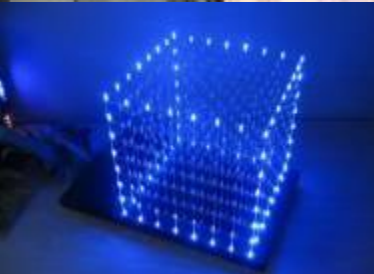
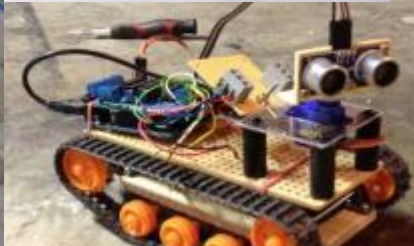
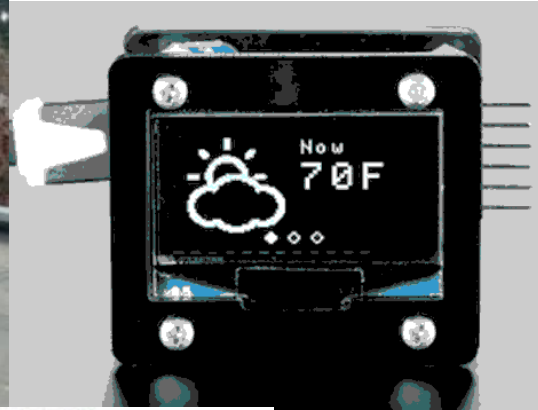
“Cheap, stripped-down microcontrollers are allowing users to pack huge amounts of computing power into tiny spaces”

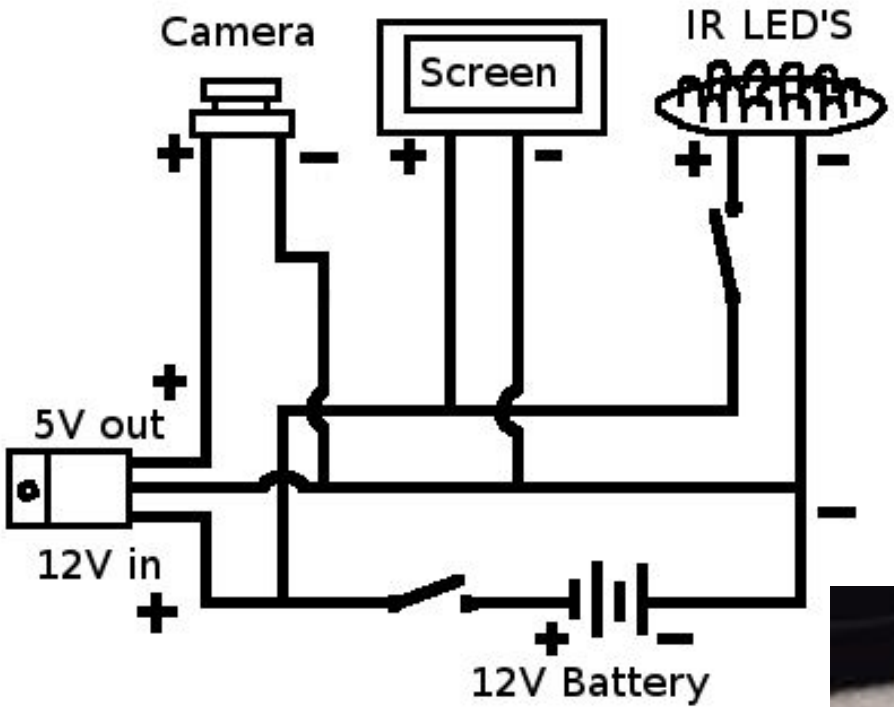


MARCH OF THE MINI-COMPUTERS

The number of papers including the terms 'Raspberry Pi' or 'Arduino' has been rising in databases of scientific publications.







DIY Infrared Night Vision Device by [MattGyver92](#)

Our Goals

- Turtle research
- GPS loggers are expensive, want alternative
- Create community
- Interdisciplinary student research

Turtle Research

Wanted: GPS logger for box turtle research

- Affordable
- Easy-to-use
- Small footprint (e.g., <5% of the body mass)



Carapace Length: ~11 – 15 cm

Weight: \bar{x} ~ 600g

Commercially available products

Desert Star Systems LLC

18g \$950

17g \$990

25g \$900

TinyTag

Custom 3D Printed Endcaps Available

Advertisement for TinyTag featuring three models of tags (18g, 17g, 25g) and a penguin. The background shows a blue sky with clouds and a moon.



e-obs bird solar tag

e.g. Bird Solar 15g
(15g, 230mAh battery & easy 100fixes/day)

GPS

- energy informed pattern
- regular short 1Hz GPS series
- continuous 1Hz GPS @fullBat

Acceleration

short raw data sequences with up to 100Hz every few minutes

UHF communication & pinger

Informational card for e-obs bird solar tag, featuring a photograph of the tag and technical specifications.

Create a community



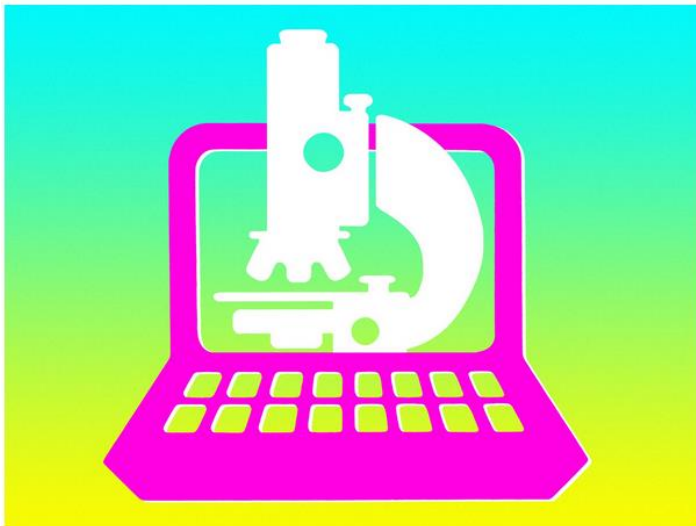
Research possibilities

- Interdisciplinary research

wired.com

Want to Make It as a Biologist? Better Learn to Code

Author: *Emily Dreyfuss. Emily Dreyfuss Science*



Evan Mills



Research possibilities

- Interdisciplinary research
- Otherwise cost-prohibitive



vs



Soldering Equipment



Soldering Iron



Lead-free solder



Brass Sponge



Third hand

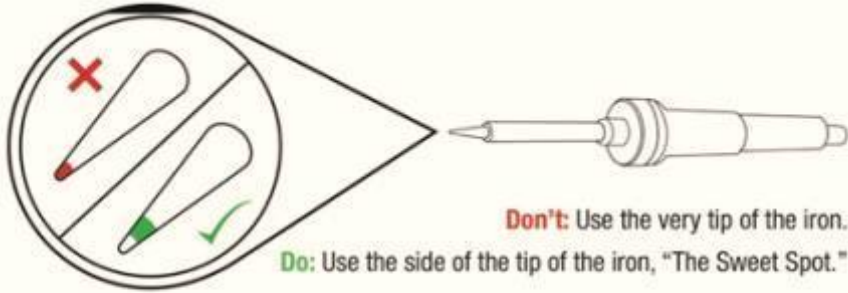


Desoldering braid



Tip cleaner

Soldering Basics



Do: Touch the iron to the component leg and metal ring at the same time.



Do: While continuing to hold the iron in contact with the leg and metal ring, feed solder into the joint.



Don't: Glob the solder straight onto the iron and try to apply the solder with the iron.



Do: Use a sponge to clean your iron whenever black oxidization builds up on the tip.



A

Solder flows around the leg and fills the hole - forming a volcano-shaped mound of solder.



B

Error: Solder balls up on the leg, not connecting the leg to the metal ring.
Solution: Add flux, then touch up with iron.



C

Error: Bad Connection (i.e. it doesn't look like a volcano)
Solution: Flux then add solder.



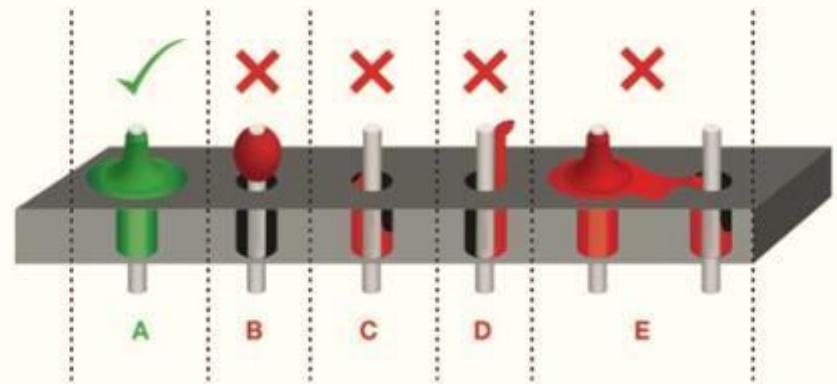
D

Error: Bad Connection...and ugly...oh so ugly.
Solution: Flux then add solder.

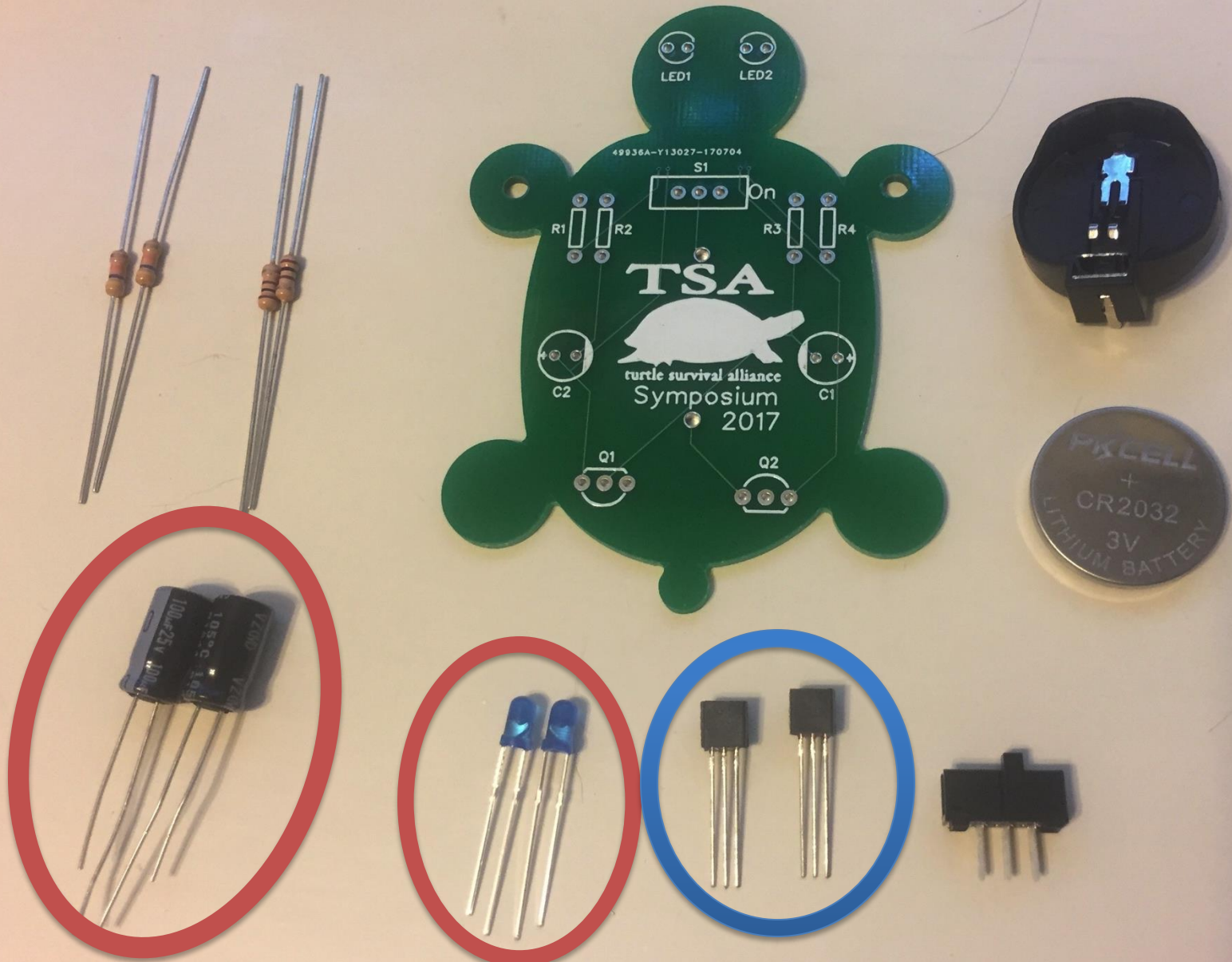


E

Error: Too much solder connecting adjacent legs (aka a solder jumper).
Solution: Wick off excess solder.



Soldering Exercise





Major Components



Arduino Pro mini microcontroller



On/Off switch



Transistor



56 Channel GPS receiver



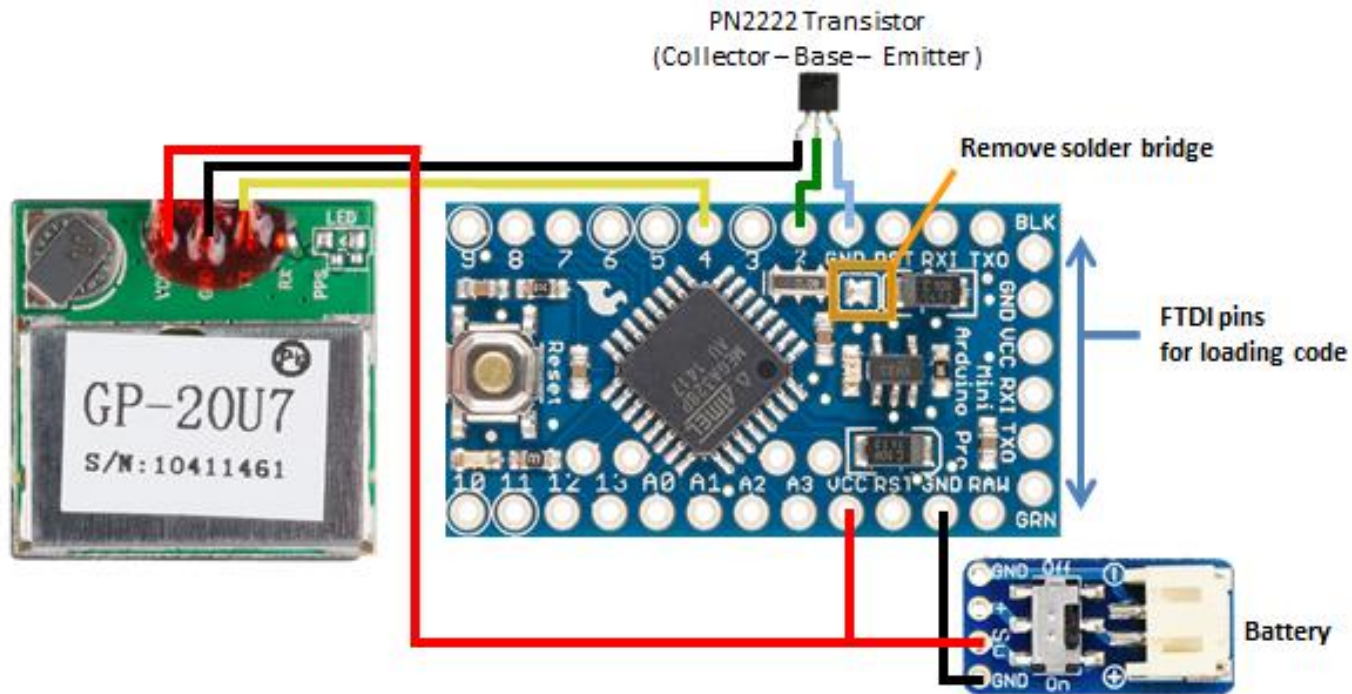
Hook-up wire (solid)



Lipo battery

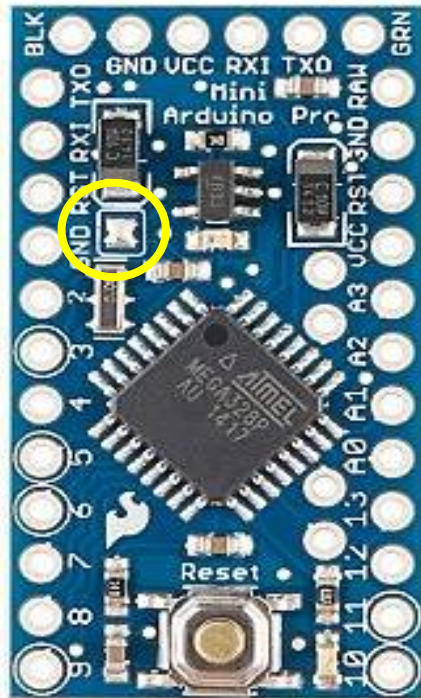
GPS unit assembly

Work in pairs

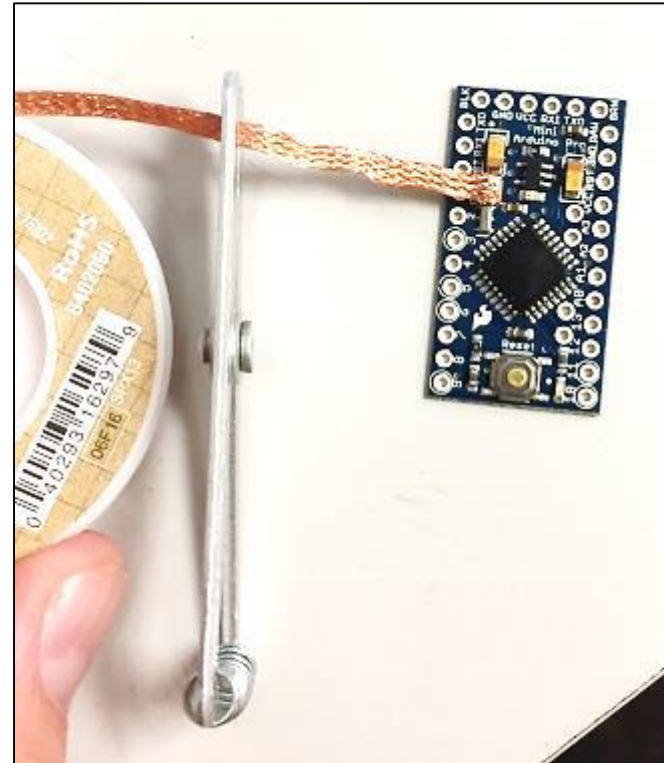
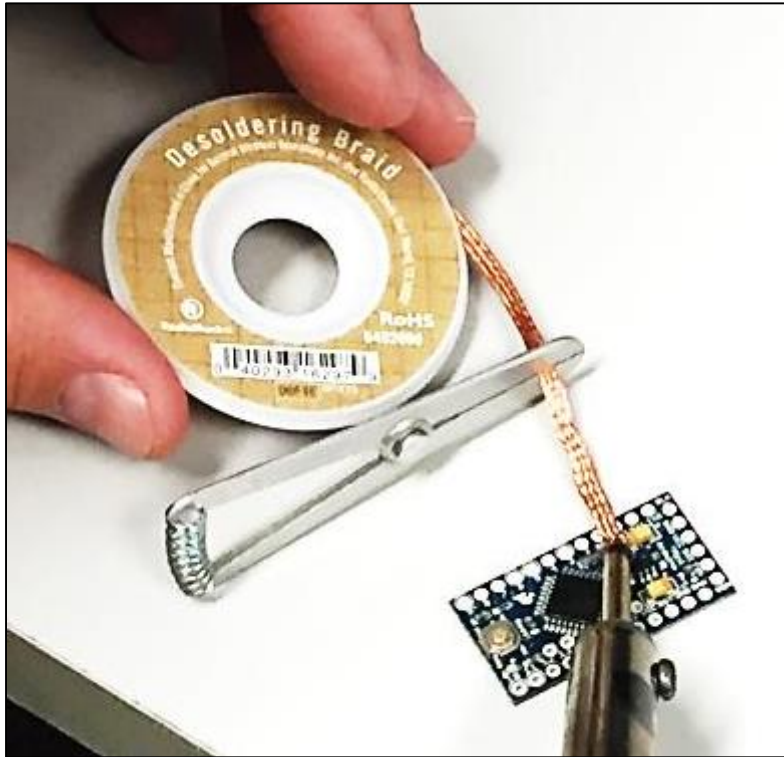


Microcontroller prep

Remove solder bridge to power regulator



Removing the solder bridge



Cut wires

Practice stripping (wires) on 22 and 28 AWG

Lengths:

Jumpers: 2.5-3.0 cm

Trim GPS wires

Black: 3 cm

Red: 3 cm

White: 4 cm



Solder jumpers to switch

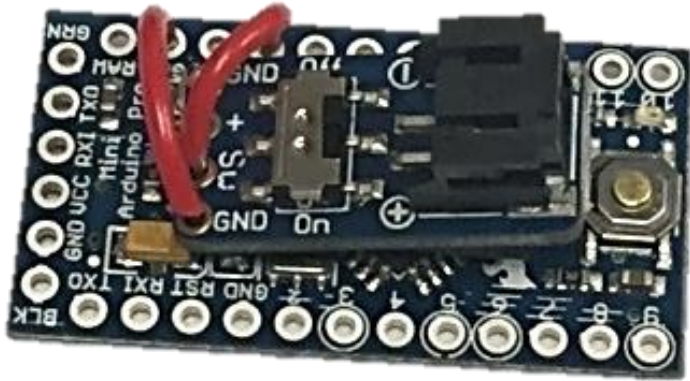
Top



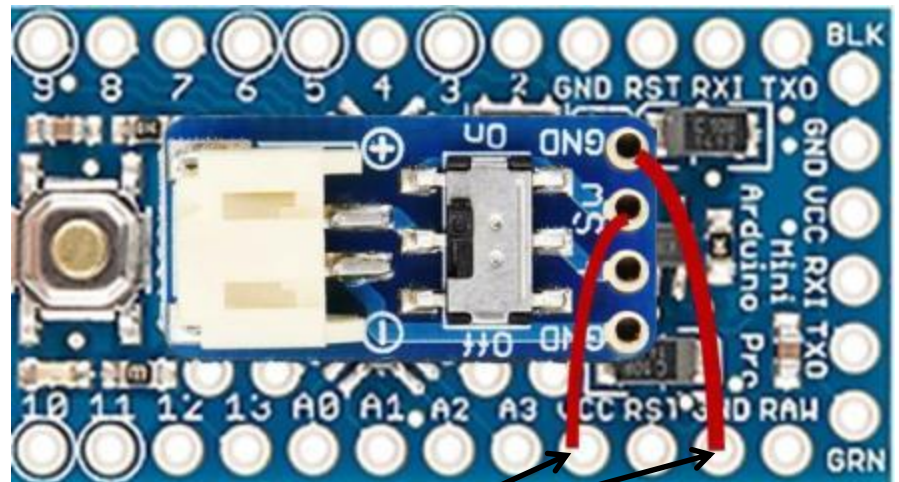
Bottom



Attach switch to Pro Mini

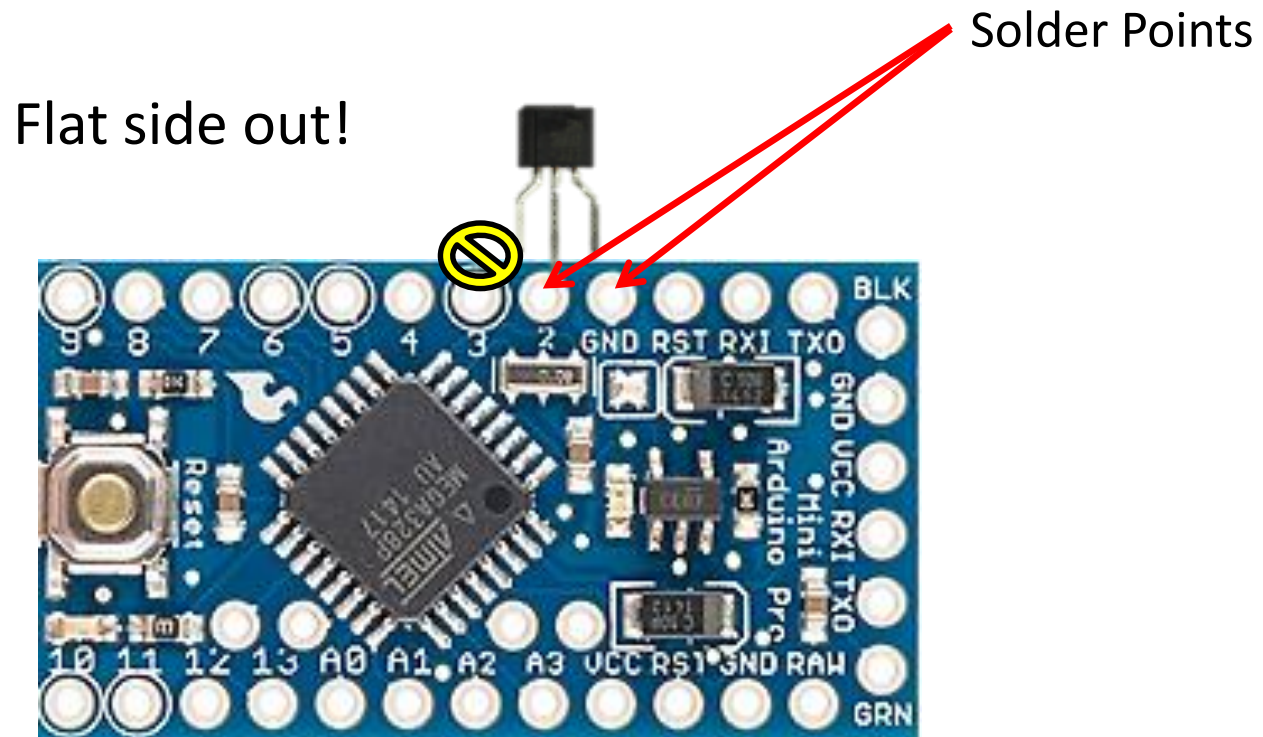


Hot glue or electrical tape

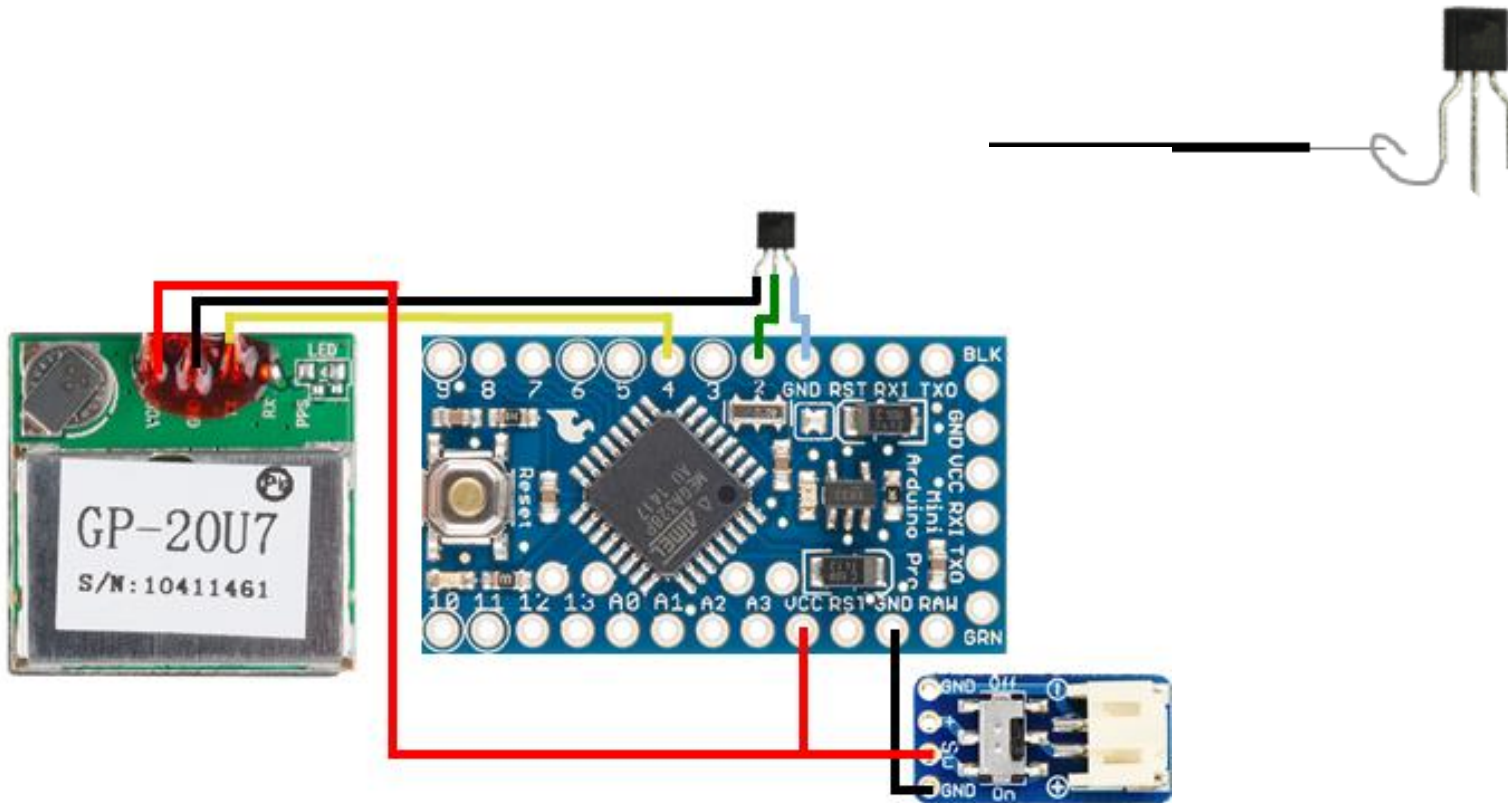


Solder points

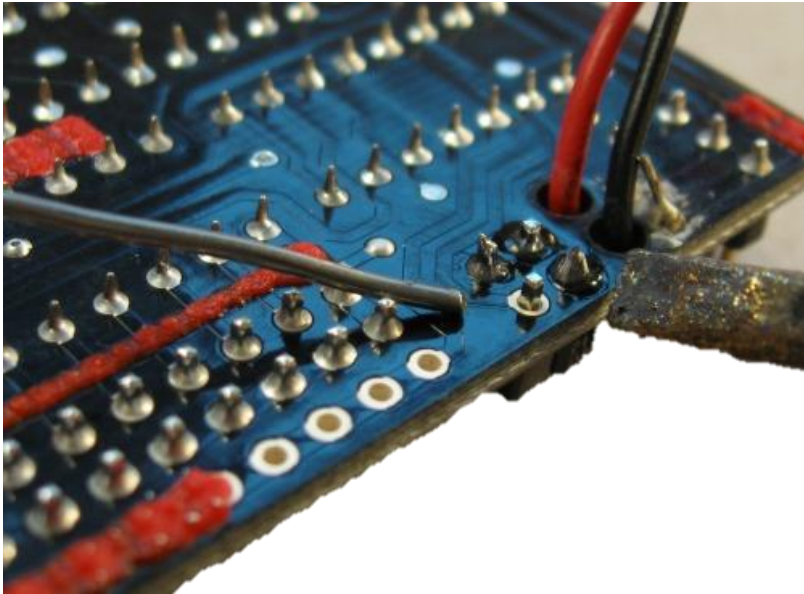
Solder transistor



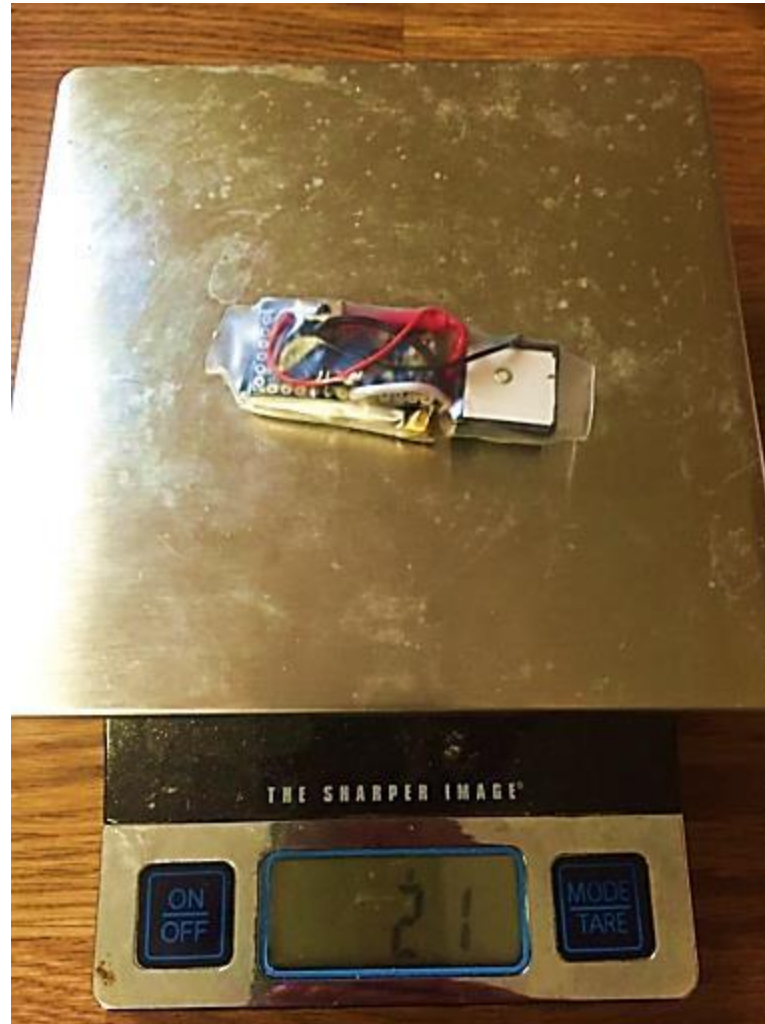
Solder GPS wires



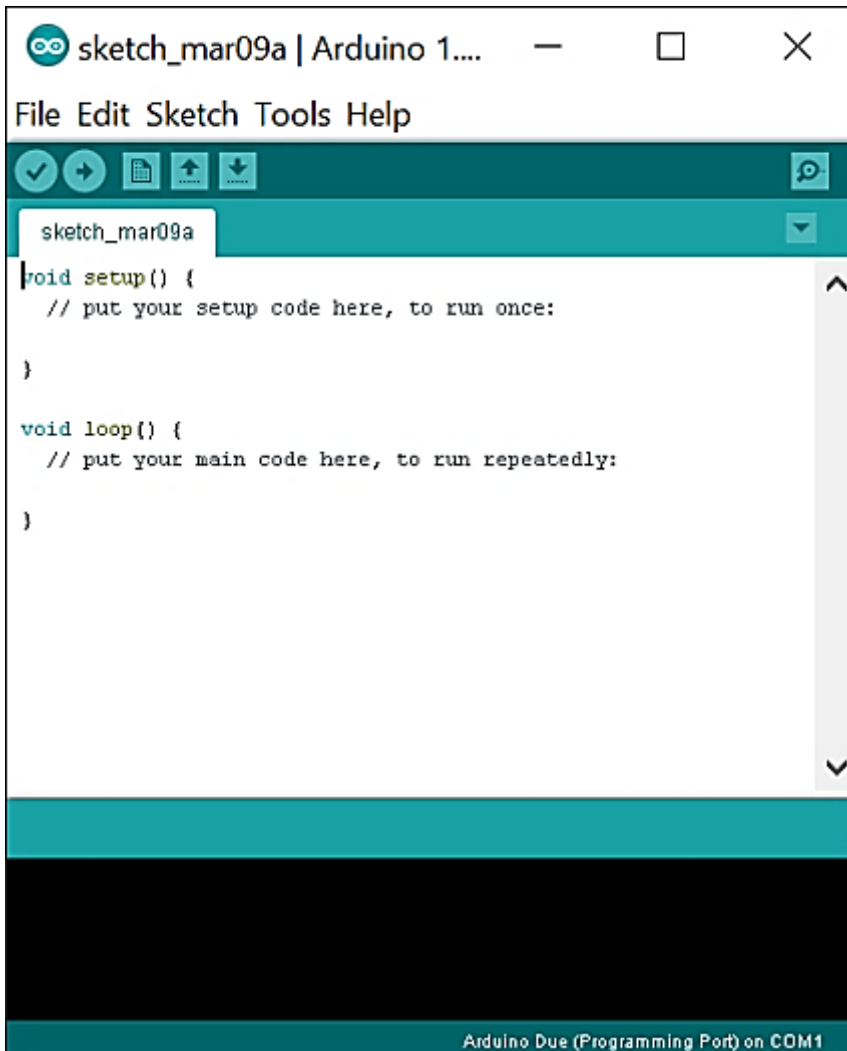
Dremel® pins on back



This is what it should look like...



Arduino IDE



Load blink sketch to
make sure it works

Install libraries:

Tiny GPS++

LowPower

EEPROMex

Load code

```
1  /*****  
2  *      Code for a turtle GPS receiver and logger  
3  *      March 2016  
4  *      P. Cain, Indiana State University, Terre Haute, IN  
5  *      -and-  
6  *      M. Cross, Bowling Green State University, Bowling Green, OH  
7  *  
8  * Microcontroller is the Arduino Pro Mini 3v3: https://www.sparkfun.com/products/11114  
9  * GPS Receiver: https://www.sparkfun.com/products/13740  
10 *  
11 * Wiring:  
12 * -GPS-  
13 * Power pin to 3v3  
14 * GND pin transistor to GND  
15 * Tx pin to D4  
16 * PN2222 Transistor: base (middle pin) to Pin D2  
17 *  
18 *****/  
19  
20 // Libraries and links to download.  
21 //For info on installing Arduino libraries, see https://www.arduino.cc/en/Guide/Libraries  
22 #include <SoftwareSerial.h>  
23 #include <TinyGPS++.h>    // http://arduiniana.org/libraries/tinygpsplus/  
24 #include <LowPower.h>    // https://github.com/rockscream/Low-Power  
25 #include <EEPROMex.h>    // http://thijs.elenbaas.net/2012/07/extended-eepron-library-for-arduino  
26  
27 /*****( USER DEFINED VARIABLES )*****/  
28 Two variables to define are how long the GPS will stay on to acquire a signal (stay_on). Tradeoff GPS  
29 up time with battery life. The other variable is how often the unit wakes and logs data (logger_interval.  
30 This occurs in hour intervals, starting on power up. Twelve hours will give you two points a day, eight  
31 will give you three, etc. */  
32  
33 // how long will GPS receiver stay on  
34 int stay_on = 2; /*minutes*/  
35  
36 // how long is the data logging interval? (time between readings)  
37 int logger_interval = 5; /*hours*/  
38 /*****/  
39  
40  
41 uint32_t feedDuration = stay_on * 60000; //time GPS is awake and feeding data to variable containers  
42 uint32_t sleepInterval = (logger_interval * 60 * 60) / 8; //raw number of cycles of watchdog interrupt  
43  
44 static const int RXPin = 4, TXPin = 3;  
45 static const uint32_t GPSSbaud = 9600;
```



Code

```
1  /*****
2  *           Code for a turtle GPS receiver and Logger
3  *           March 2016
4  *           P. Cain, Indiana State University, Terre Haute, IN
5  *           -and-
6  *           M. Cross, Bowling Green State University, Bowling Green, OH
7  *
8  * Microcontroller is the Arduino Pro Mini 3v3: https://www.sparkfun.com/products/11114
9  * GPS Receiver: https://www.sparkfun.com/products/13740
10 *
11 * Wiring:
12 * -GPS-
13 * Power pin to 3v3
14 * GND pin transistor to GND
15 * Tx pin to D4
16 * PN2222 Transistor: base (middle pin) to Pin D2
17 *
18 *****/
19
20 // Libraries and Links to download.
21 //For info on installing Arduino libraries, see https://www.arduino.cc/en/Guide/Libraries
22 #include <SoftwareSerial.h>
23 #include <TinyGPS++.h> // http://arduiniiana.org/libraries/tinygpsplus/
24 #include <LowPower.h> // https://github.com/roocketscream/Low-Power
25 #include <EEPROMex.h> // http://thijs.elenbaas.net/2012/07/extended-eepron-library-for-arduino
26
27 /*****( USER DEFINED VARIABLES )*****/
28 Two variables to define are how long the GPS will stay on to acquire a signal (stay_on). Tradeoff GPS
29 up time with battery life. The other variable is how often the unit wakes and logs data (logger_interval.
30 This occurs in hour intervals, starting on power up. Twelve hours will give you two points a day, eight
31 will give you three, etc. */
32
33 // how long will GPS receiver stay on
34 int stay_on = 2; /*minutes*/
35
36 // how long is the data logging interval? (time between readings)
37 int logger_interval = 5; /*hours*/
38 /*****/
39
40
41 uint32_t feedDuration = stay_on * 60000; //time GPS is awake and feeding data to variable containers
42 uint32_t sleepInterval = (logger_interval * 60 * 60) / 8; //raw number of cycles of watchdog interrupt
43
44 static const int RXPin = 4, TXPin = 3;
45 static const uint32_t GPSSBaud = 9600;
```

To Change Recording Frequency

```
20
27 /*****( USER DEFINED VARIABLES )*****/
28 Two variables to define are how long the GPS will stay on to acquire a signal (stay_on). Tradeoff GPS
29 up time with battery life. The other variable is how often the unit wakes and logs data (logger_interval.
30 This occurs in hour intervals, starting on power up. Twelve hours will give you two points a day, eight
31 will give you three, etc. */
32
33 // how long will GPS receiver stay on
34 int stay_on = 2; /*minutes*/
35
36 // how long is the data logging interval? (time between readings)
37 int logger_interval = 5; /*hours*/
38 /*****/
```

You can change how long the unit attempts to get a fix

This is where you can change the time between readings

For now set the interval to record once every minute or 0.017

Remember Adjusting these parameters will affect battery life and the unit has a limited amount of memory (83 locations).

Break!!!

Seneca Park Zoo



Data download

Bring up GPSlogger_ReadClear sketch.

Connect FTDI connector.

Load sketch to logger.

Type "R" for read.

Troubleshoot.

The screenshot shows the Arduino IDE interface. The left pane displays the sketch code for 'GPSlogger_ReadClear'. The right pane shows the serial monitor output, which includes a list of GPS coordinates and timestamps, indicating that the sketch is successfully reading data from the EEPROM.

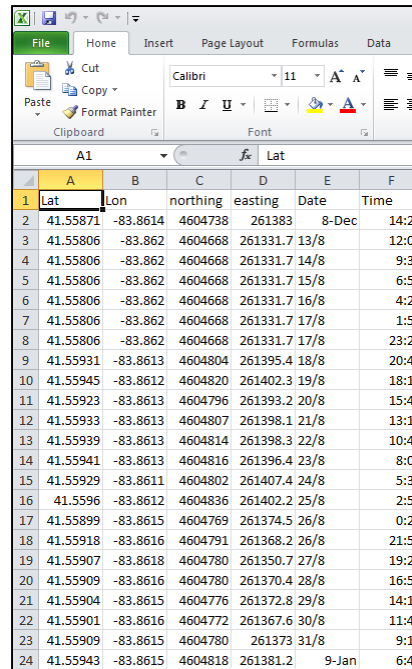
```
1 /*
2 This sketch is used to manage data stored to the EEPROM of an Arduino Pro Mini Atmega
3 chip via GPSlogger sketch.
4
5 Currently has three functions:
6 1) (R) Read all non-zero values from address 0
7 2) (C) Clear EEPROM (write zeros to all addresses)
8 3) (A) Check last address for address value last written to
9 4) (Z) Forces printing of every address from 0 to address specified
10
11 */
12
13 #include <EEPROM.h>
14
15 int address = 0;
16
17 double latOut = 0;
18 double lonOut = 0;
19 byte dayOut = 0;
20 byte monthOut = 0;
21 byte hourOut = 0;
22 byte minuteOut = 0;
23 byte secOut = 0;
24
25 int addressDouble = sizeof(double);
26 int addressByte = sizeof(byte);
27 const int memBase = 0;
28
29 // Flag for Read loop
30 int x = 0;
31
32 #define LED 13
33
34 void setup() {
35   Serial.begin(115200);
36
37   EEPROM.setMemPool(memBase, EEPROMSIZELNO);
38   const int maxAllowedWrites = 1024;
39   EEPROM.setMaxAllowedWrites(maxAllowedWrites);
40
41   Done uploading.
42
43   Sketch uses 5574 bytes (18%) of program storage space. Maximum is 30720 bytes.
44   Global variables use 640 bytes (31%) of dynamic memory, leaving 3408 bytes for local variables. Maximum is
45   32768 bytes.
46
47   Tue Mar 14 21:52 Pat Cain /dev/cu.usbserial-A603H8A3
48
49   R
50
51   Please type C to clear EEPROM, R to Read, A to print last address used
52
53   ***** COPY Begin *****
54   Lat, Lon, Date, Time
55   41.3709367, -83.6179800, 15/3, 01:40
56   41.3709322, -83.617897, 15/3, 01:41
57   41.3709441, -83.617912, 15/3, 01:42
58   41.370887, -83.617912, 15/3, 01:43
59   41.370899, -83.617889, 15/3, 01:44
60   41.370918, -83.617912, 15/3, 01:45
61   41.370880, -83.617904, 15/3, 01:46
62   41.370910, -83.617919, 15/3, 01:47
63   41.370914, -83.617897, 15/3, 01:48
64   ***** Finished *****
65
66   Please type C to clear EEPROM, R to Read, A to print last address used
```

Data Visualization

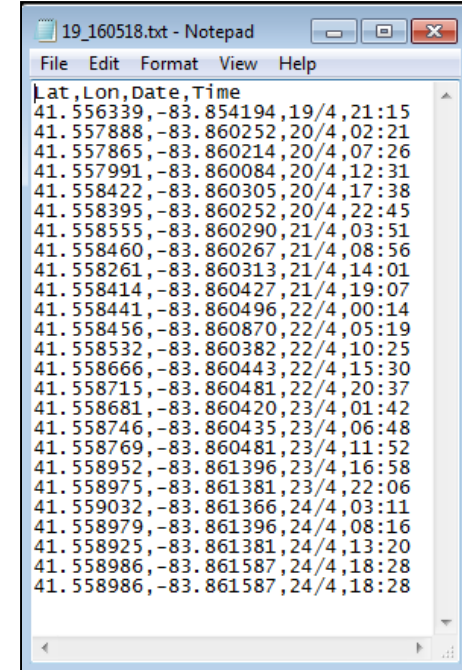
Copy data into notepad

Save as .csv file

Import...

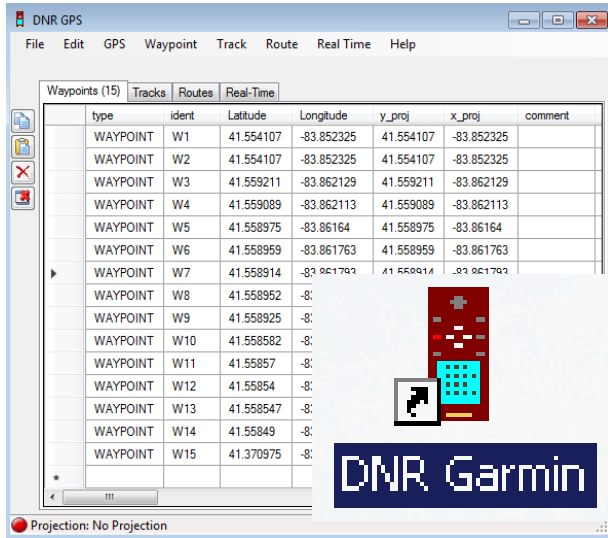


	A	B	C	D	E	F
1	Lat	Lon	northing	easting	Date	Time
2	41.55871	-83.8614	4604738	261383	8-Dec	14:29
3	41.55806	-83.862	4604668	261331.7	13/8	12:00
4	41.55806	-83.862	4604668	261331.7	14/8	9:30
5	41.55806	-83.862	4604668	261331.7	15/8	6:58
6	41.55806	-83.862	4604668	261331.7	16/8	4:26
7	41.55806	-83.862	4604668	261331.7	17/8	1:54
8	41.55806	-83.862	4604668	261331.7	17/8	23:22
9	41.55931	-83.8613	4604804	261395.4	18/8	20:49
10	41.55945	-83.8612	4604820	261402.3	19/8	18:18
11	41.55923	-83.8613	4604796	261393.2	20/8	15:47
12	41.55933	-83.8613	4604807	261398.1	21/8	13:15
13	41.55939	-83.8613	4604814	261398.3	22/8	10:40
14	41.55941	-83.8613	4604816	261396.4	23/8	8:05
15	41.55929	-83.8611	4604802	261407.4	24/8	5:31
16	41.5596	-83.8612	4604836	261402.2	25/8	2:58
17	41.55899	-83.8615	4604769	261374.5	26/8	0:26
18	41.55918	-83.8616	4604791	261368.2	26/8	21:55
19	41.55907	-83.8618	4604780	261350.7	27/8	19:23
20	41.55909	-83.8616	4604780	261370.4	28/8	16:50
21	41.55904	-83.8615	4604776	261372.8	29/8	14:18
22	41.55901	-83.8616	4604772	261367.6	30/8	11:46
23	41.55909	-83.8615	4604780	261373	31/8	9:14
24	41.55943	-83.8615	4604818	261381.2	9-Jan	6:41



```
19_160518.txt - Notepad
File Edit Format View Help
Lat, Lon, Date, Time
41.556339, -83.854194, 19/4, 21:15
41.557888, -83.860252, 20/4, 02:21
41.557865, -83.860214, 20/4, 07:26
41.557991, -83.860084, 20/4, 12:31
41.558422, -83.860305, 20/4, 17:38
41.558395, -83.860252, 20/4, 22:45
41.558555, -83.860290, 21/4, 03:51
41.558460, -83.860267, 21/4, 08:56
41.558261, -83.860313, 21/4, 14:01
41.558414, -83.860427, 21/4, 19:07
41.558441, -83.860496, 22/4, 00:14
41.558456, -83.860870, 22/4, 05:19
41.558532, -83.860382, 22/4, 10:25
41.558666, -83.860443, 22/4, 15:30
41.558715, -83.860481, 22/4, 20:37
41.558681, -83.860420, 23/4, 01:42
41.558746, -83.860435, 23/4, 06:48
41.558769, -83.860481, 23/4, 11:52
41.558952, -83.861396, 23/4, 16:58
41.558975, -83.861381, 23/4, 22:06
41.559032, -83.861366, 24/4, 03:11
41.558979, -83.861396, 24/4, 08:16
41.558925, -83.861381, 24/4, 13:20
41.558986, -83.861587, 24/4, 18:28
41.558986, -83.861587, 24/4, 18:28
```



Data visualization



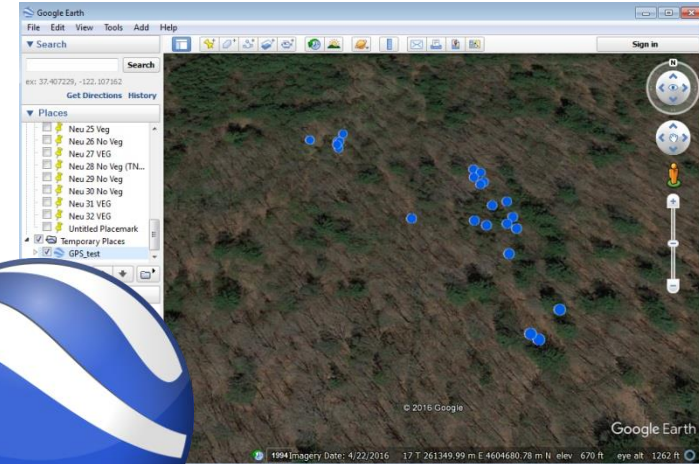
DNR GPS

Waypoints (15)	Tracks	Routes	Real-Time			
type	ident	Latitude	Longitude	y_proj	x_proj	comment
WAYPOINT	W1	41.554107	-83.852325	41.554107	-83.852325	
WAYPOINT	W2	41.554107	-83.852325	41.554107	-83.852325	
WAYPOINT	W3	41.559211	-83.862129	41.559211	-83.862129	
WAYPOINT	W4	41.559089	-83.862113	41.559089	-83.862113	
WAYPOINT	W5	41.558975	-83.86164	41.558975	-83.86164	
WAYPOINT	W6	41.558959	-83.861763	41.558959	-83.861763	
WAYPOINT	W7	41.558914	-83.861793	41.558914	-83.861793	
WAYPOINT	W8	41.558952	-83.861793	-83.861793		
WAYPOINT	W9	41.558925	-83.861793	-83.861793		
WAYPOINT	W10	41.558882	-83.861793	-83.861793		
WAYPOINT	W11	41.55857	-83.861793	-83.861793		
WAYPOINT	W12	41.55854	-83.861793	-83.861793		
WAYPOINT	W13	41.558547	-83.861793	-83.861793		
WAYPOINT	W14	41.55849	-83.861793	-83.861793		
WAYPOINT	W15	41.370975	-83.861793	-83.861793		

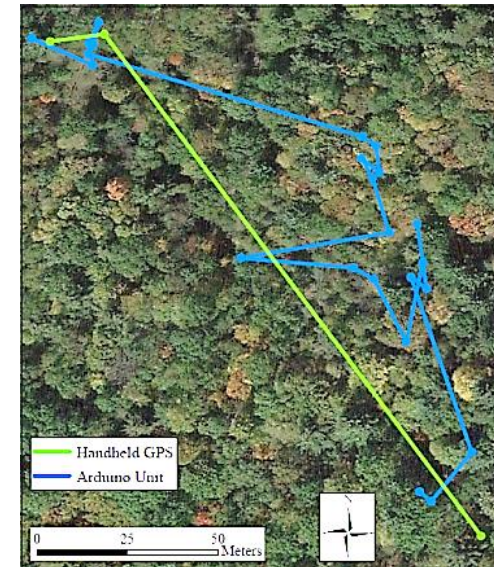
Projection: No Projection



DNR Garmin



GRASS GIS



Other DIY GPS Logger Research: As-is

Desert Star Systems LLC

18g \$950 17g \$990 25g \$900

TinyTag

Custom 3D Printed Endcaps Available

The advertisement features a large image of a King penguin with a TinyTag attached to its back. The tag is shown in a circular inset with three different models. The background is a blue sky with clouds and a small globe in the bottom left corner.



A photograph of several product cards for GPS loggers. The cards are white with blue and black text and images. One card prominently displays the following information:

e-obs Bird solar tag

e.g. Bird Solar 15g
(15g, 230mAh battery & easy 100fixes/day)

GPS

- energy informed pattern
- regular short 1Hz GPS series
- continuous 1Hz GPS @fullBat

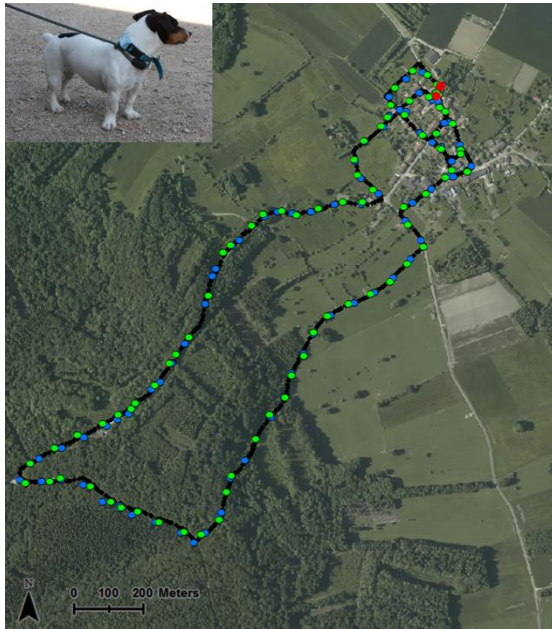
Acceleration

short raw data sequences with up to 100Hz every few minutes

UHF communication & pinger

Other cards in the background show similar information for different models.

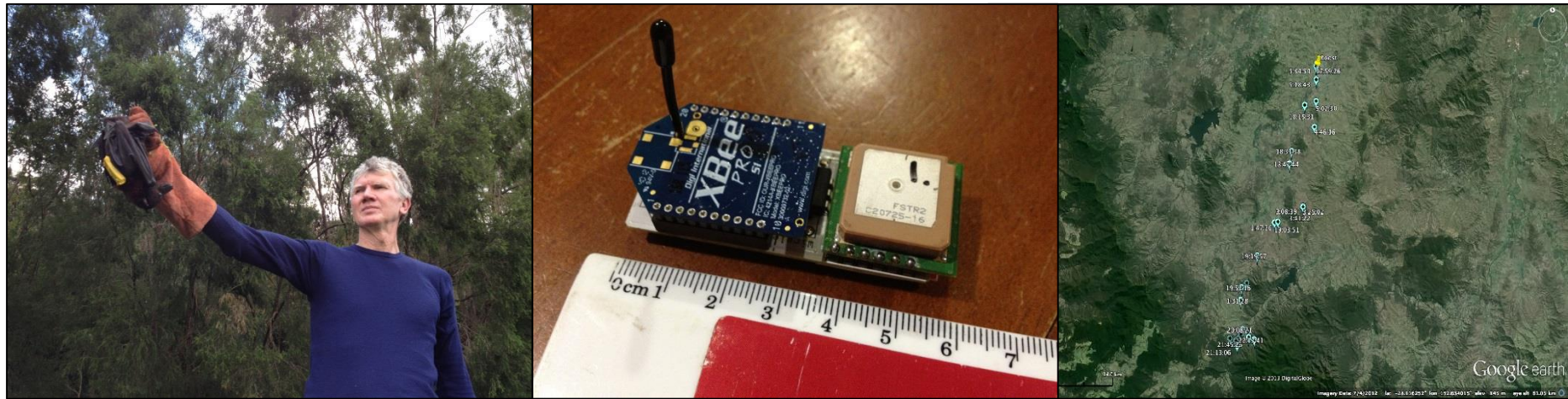
Other DIY GPS Logger Research: Modifications



Other DIY GPS Logger Research: DIY

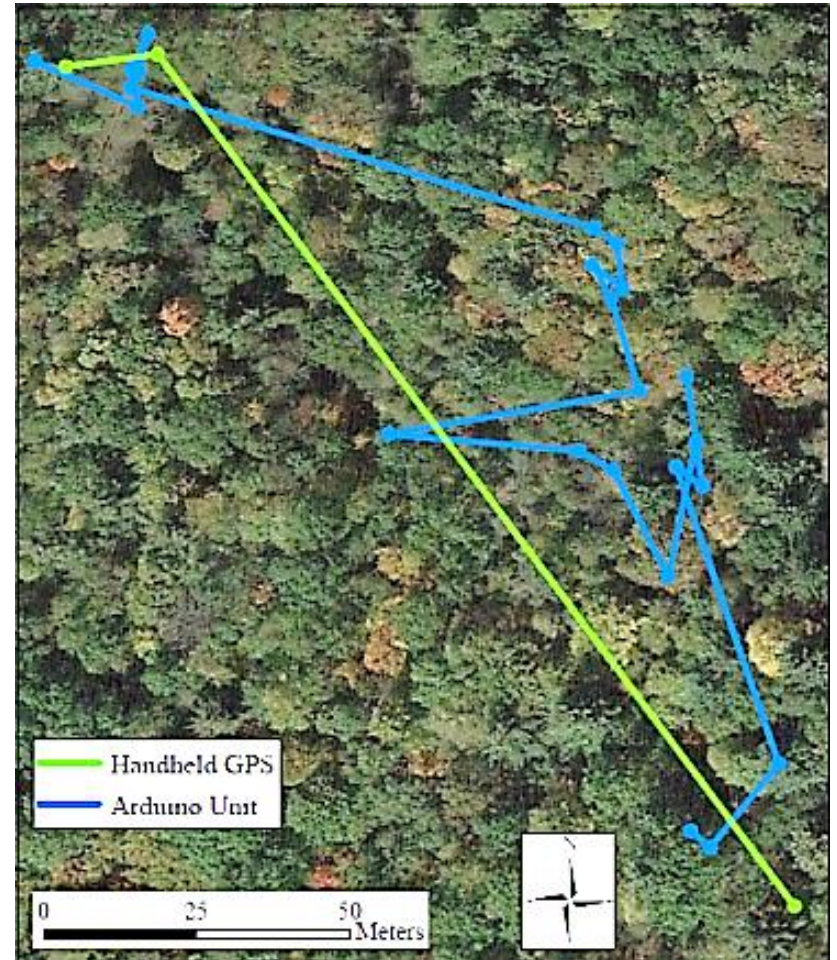
A Low-Cost GPS GSM/GPRS Telemetry System:
Performance in Stationary Field Tests and Preliminary
Data on Wild Otters (*Lutra lutra*)

Lorenzo Quaglietta^{1*}, Bruno Herlander Martins², Addy de Jongh³, António Mira^{4*}, Luigi Boitani¹

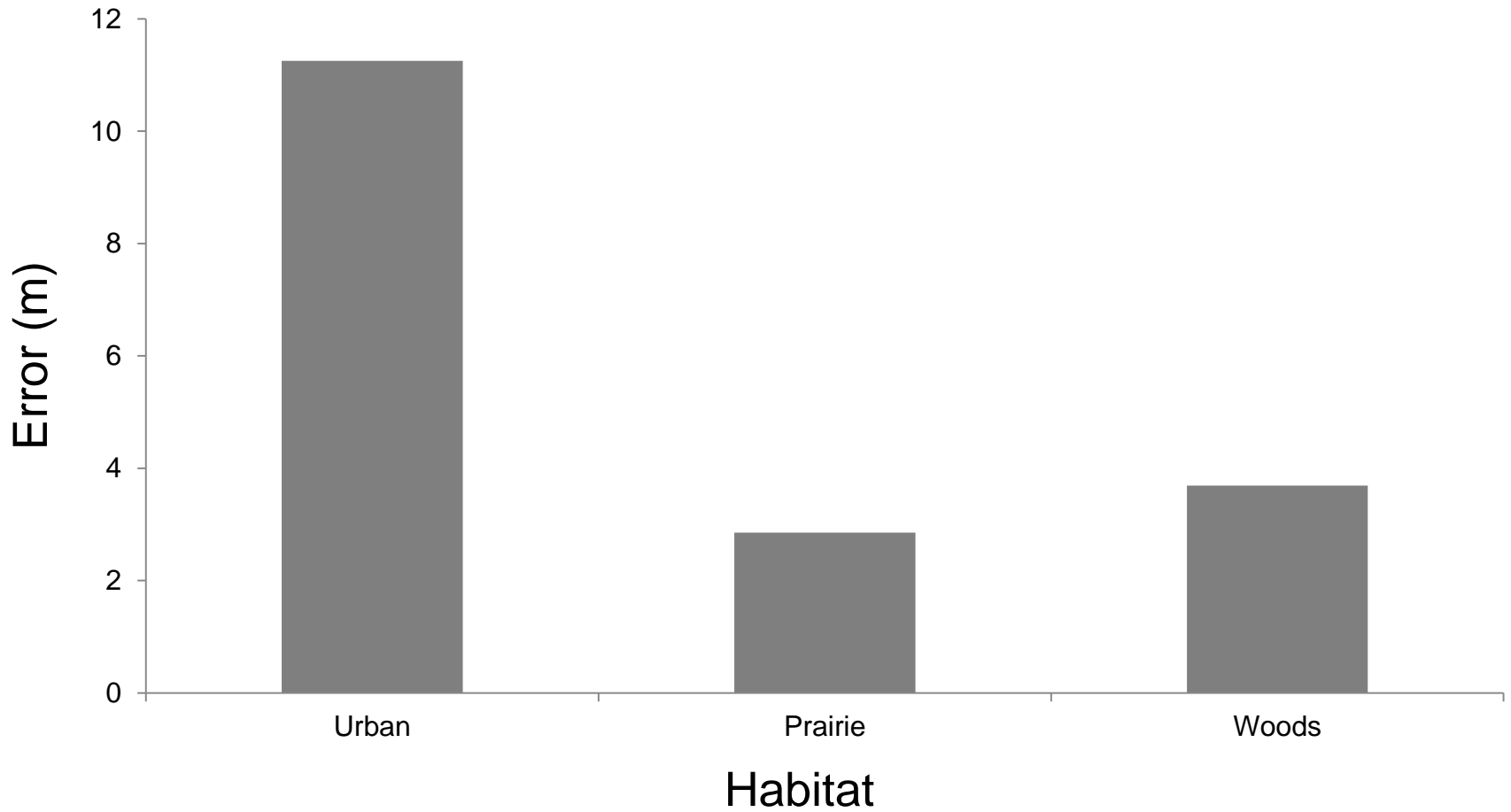


<http://fritzing.org/projects/wireless-gps-datalogger>

Turtle test



Accuracy assessment



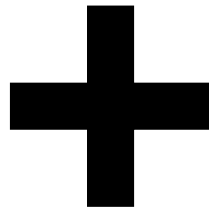
Attachment methods



Price-Rees and Shine 2011

Flesch et al. 2009

Casing options





Interdisciplinary/Outreach

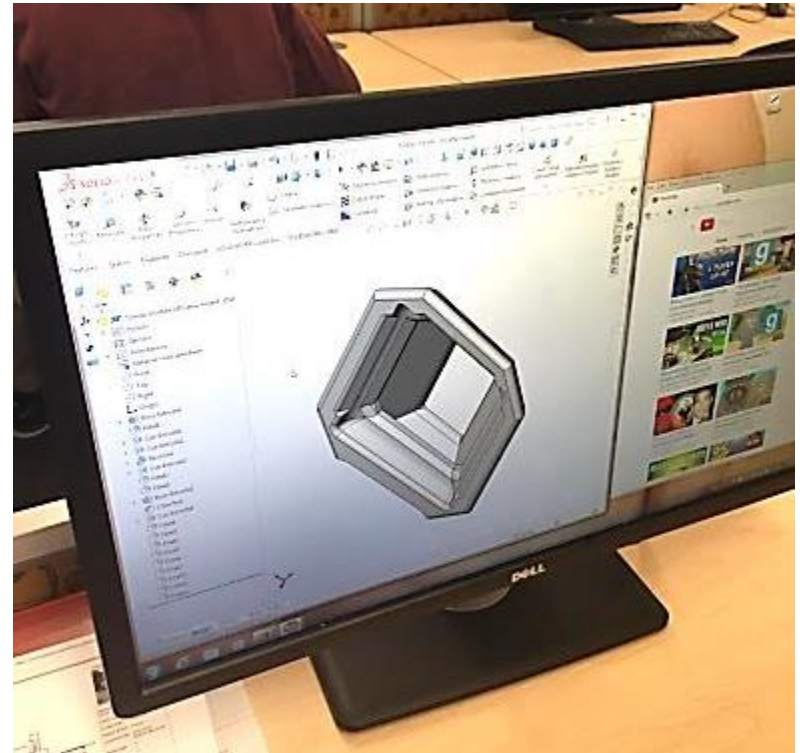
Presented high school students with a problem to solve:

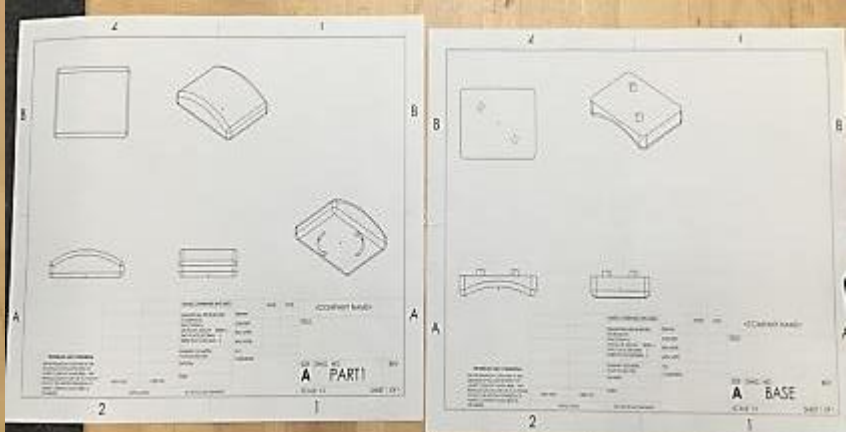
“Design a GPS logger case for box turtles”.

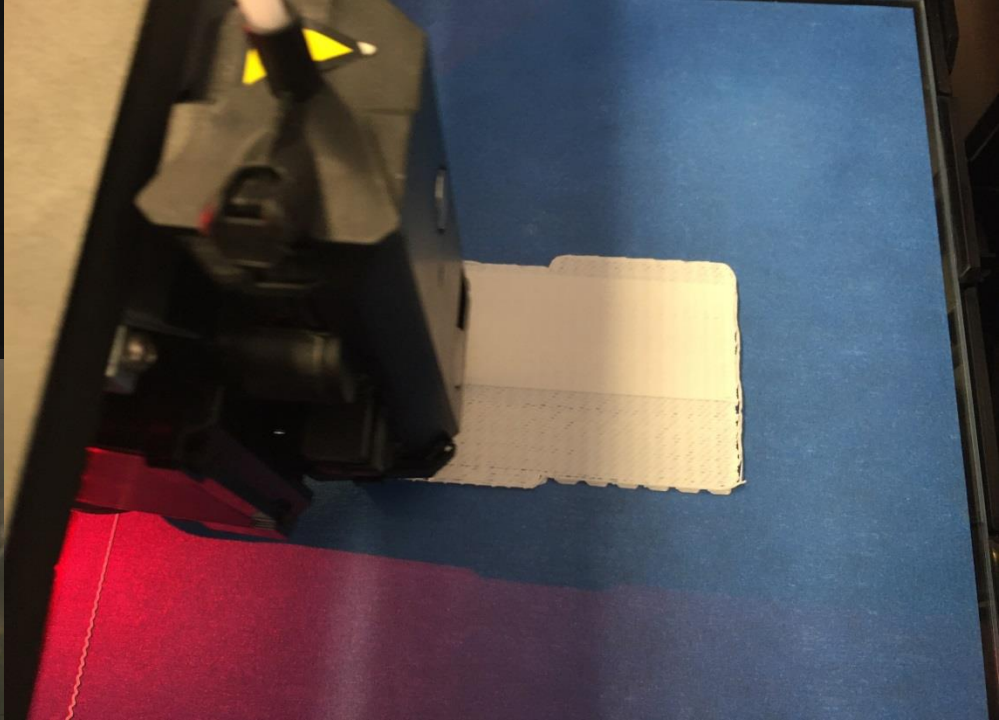
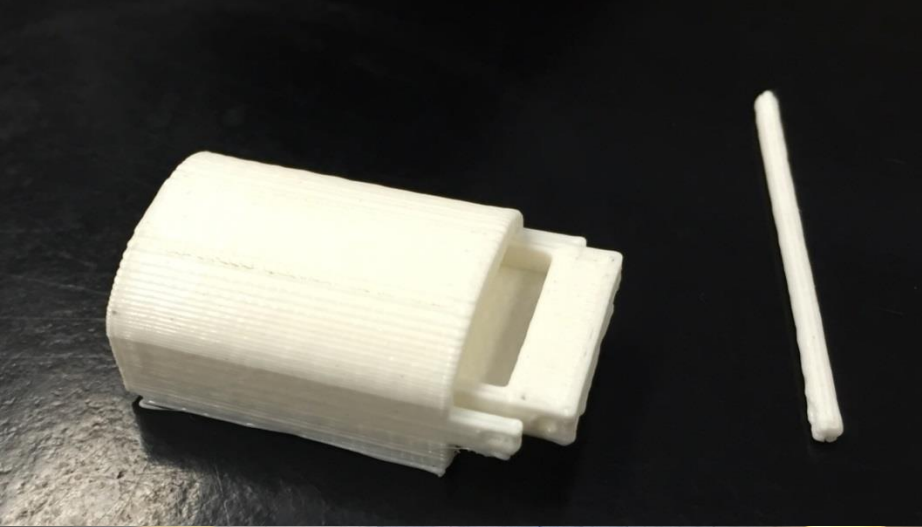
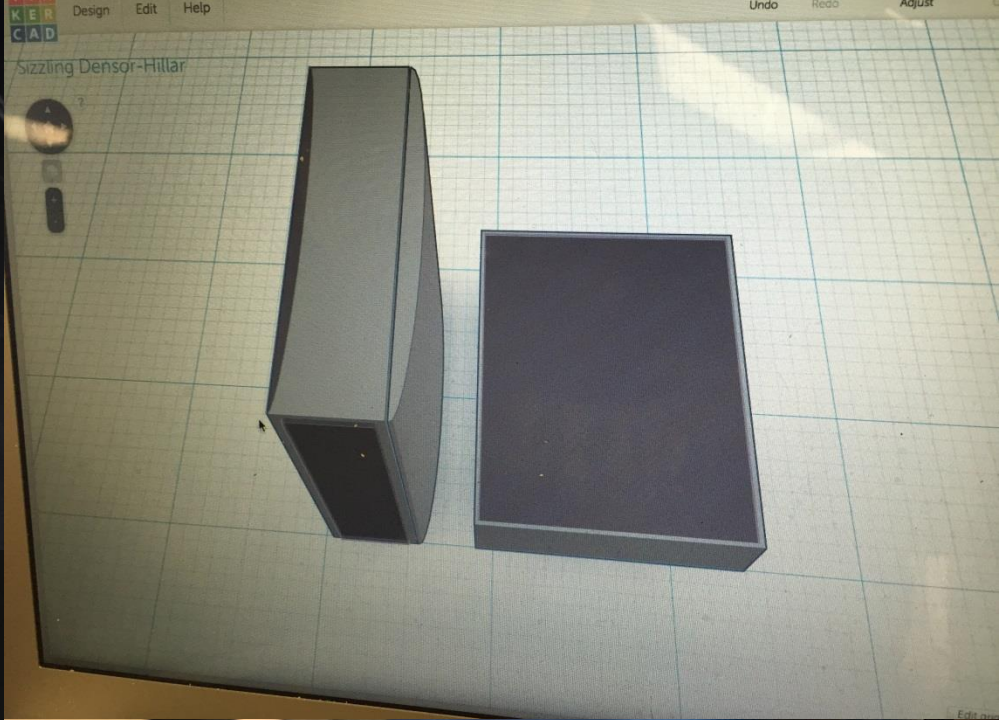
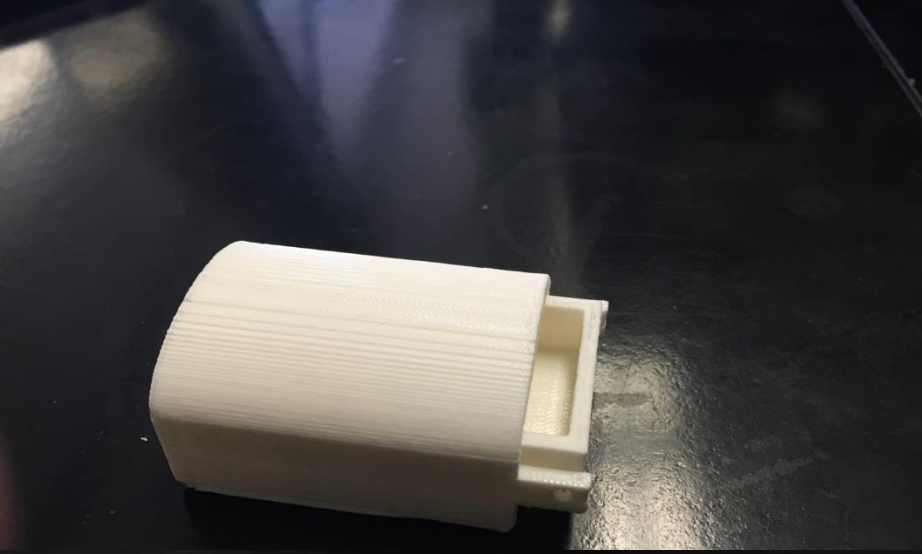
- Specs
 - GPS logger should be easily accessed
 - Should not increase the footprint of the logger by too much
 - Lightweight
- The rest was up to them



Response was incredible

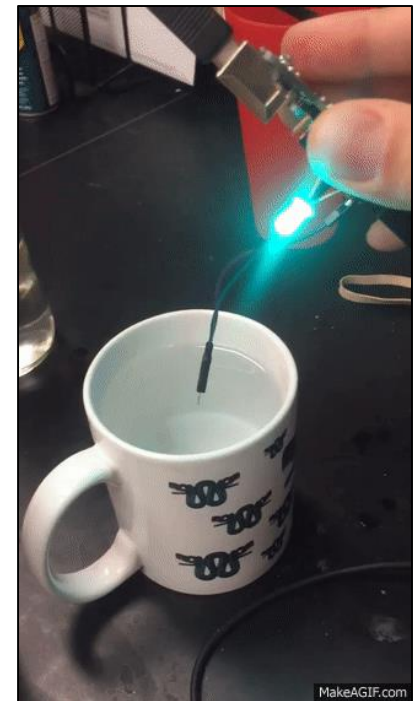
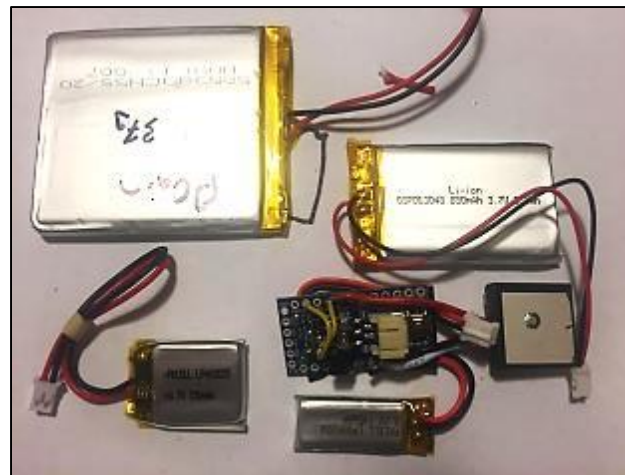




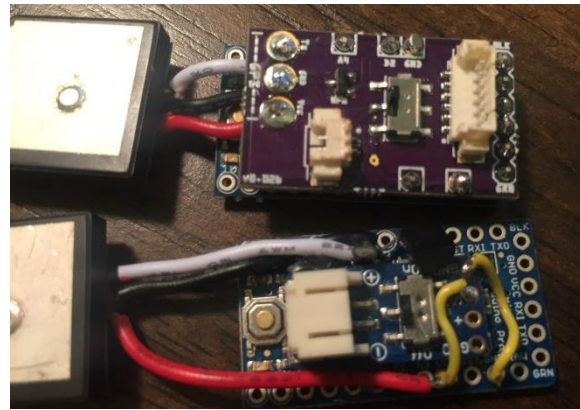
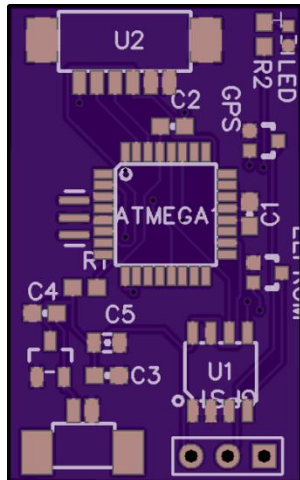
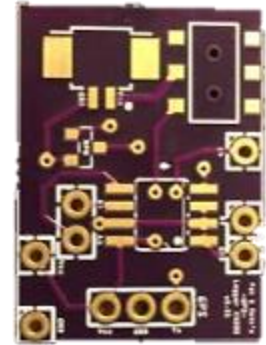
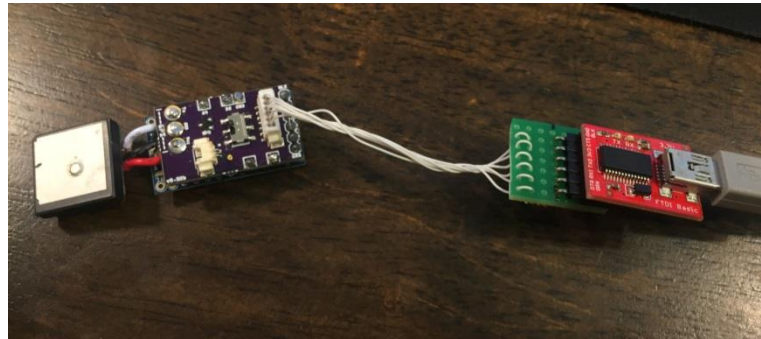
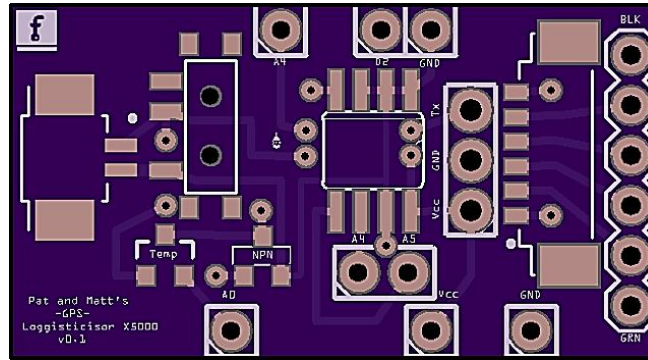
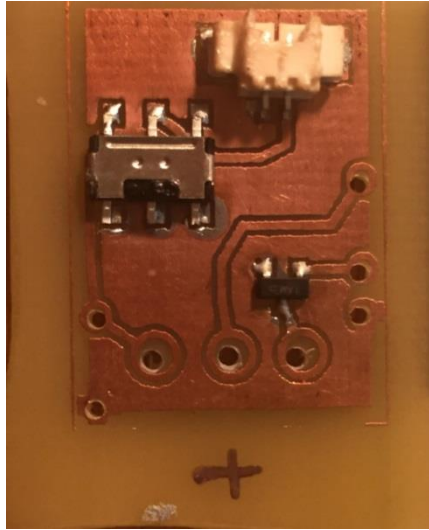
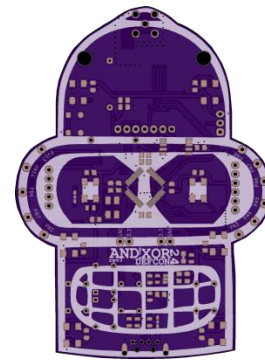


Customization

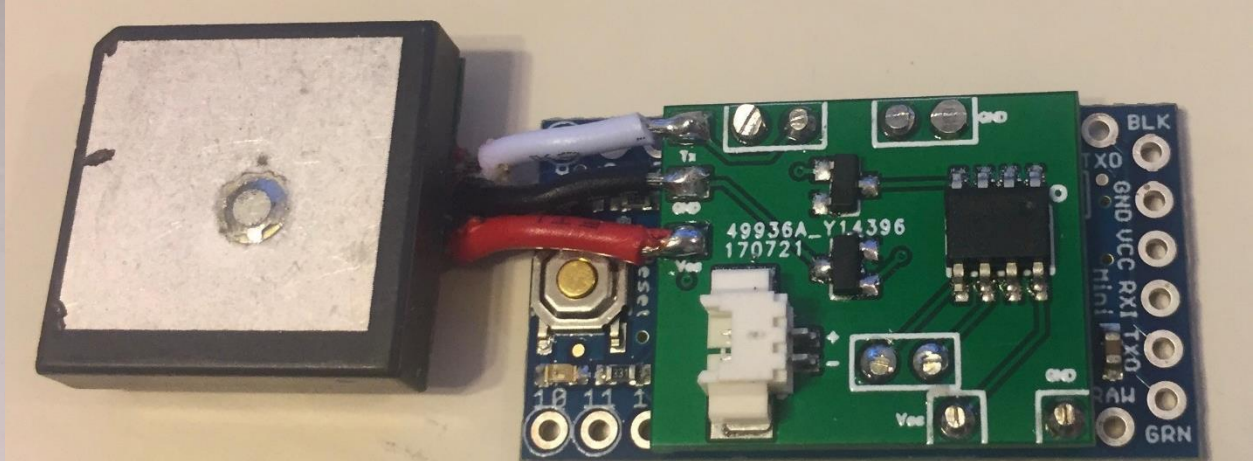
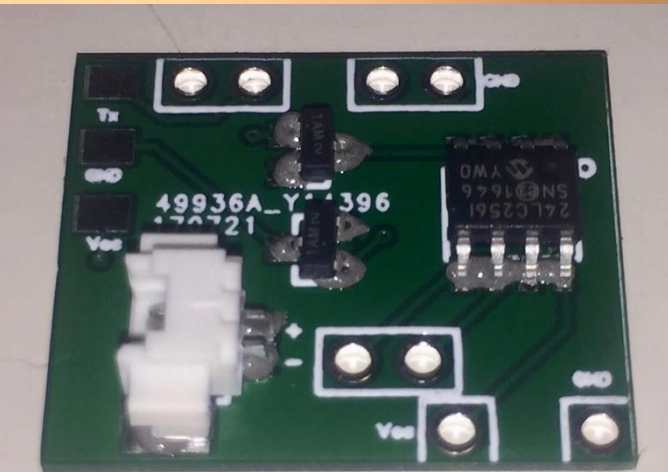
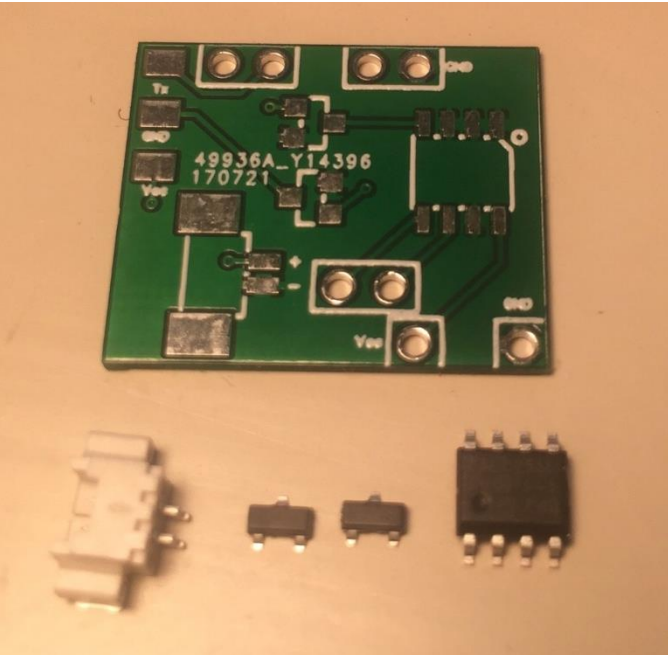
- Temperature
- Remote download
- Increased storage
- Etc...



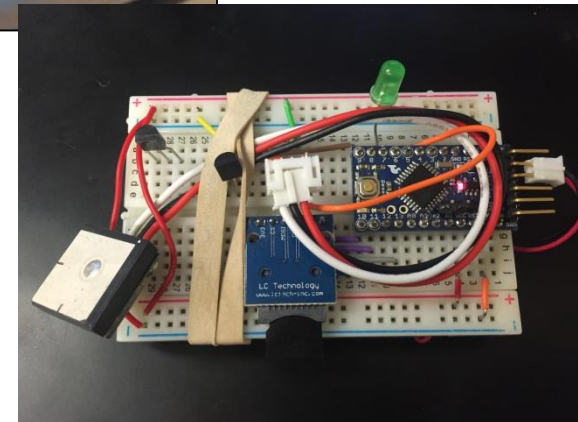
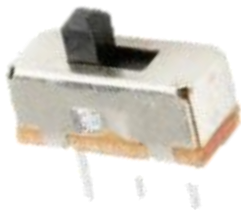
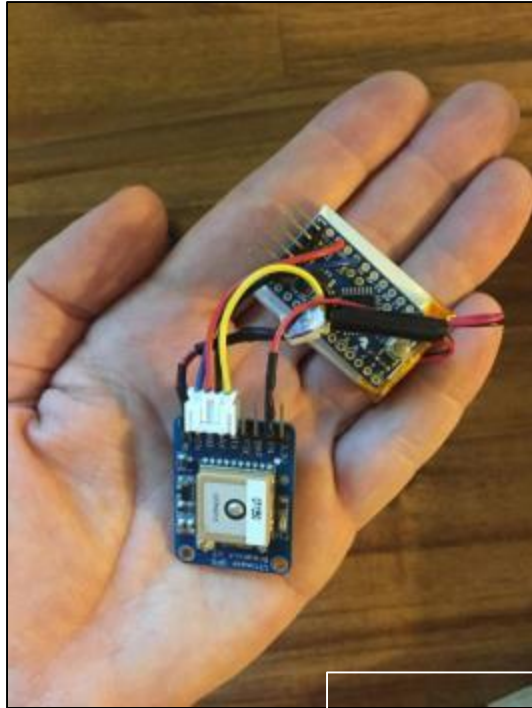
Customization: PCBs



TNG expanded memory



GPS logger evolution



Will still need a transmitter!!!



Acknowledgements

RocketScream (low power library)

Mikal Hart (TinyGPS++ library)

Thijs Elebaas (extended EEPROM library)

Sparkfun.com

Adafruit.com

Electricity



Questions?

