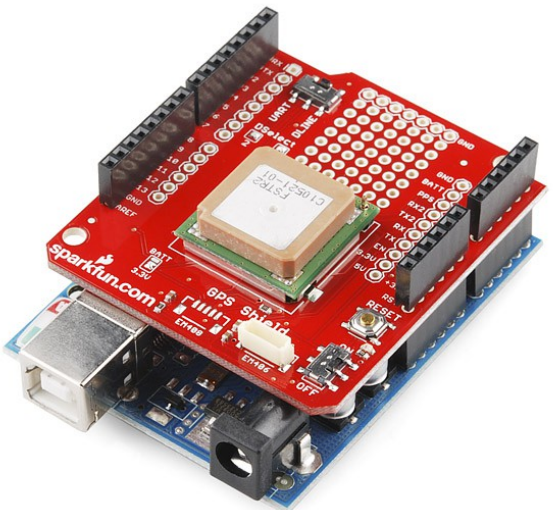
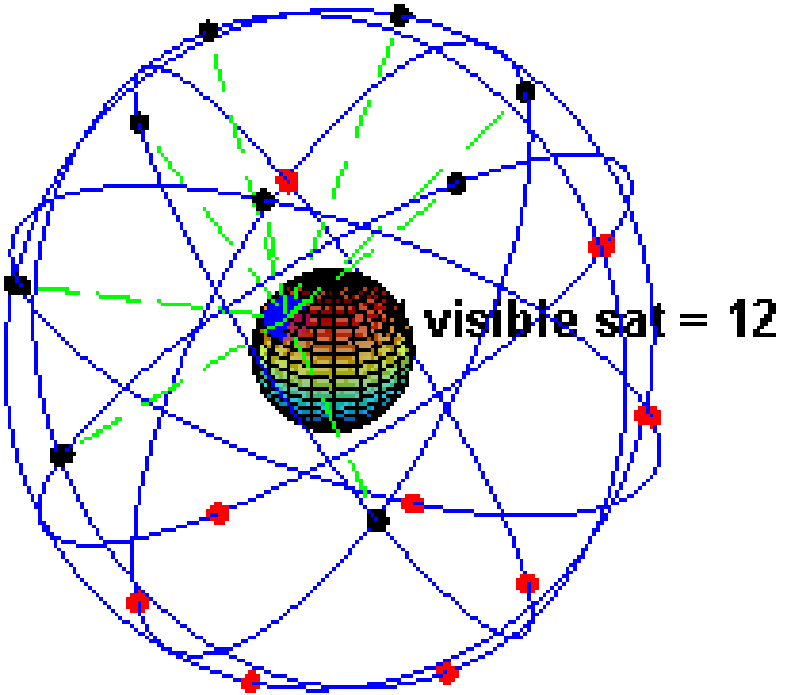
# GPS - Global Positioning System

**Mike Grusin**

**Engineer**
**Sparkfun Electronics**
**mike.grusin@sparkfun.com**

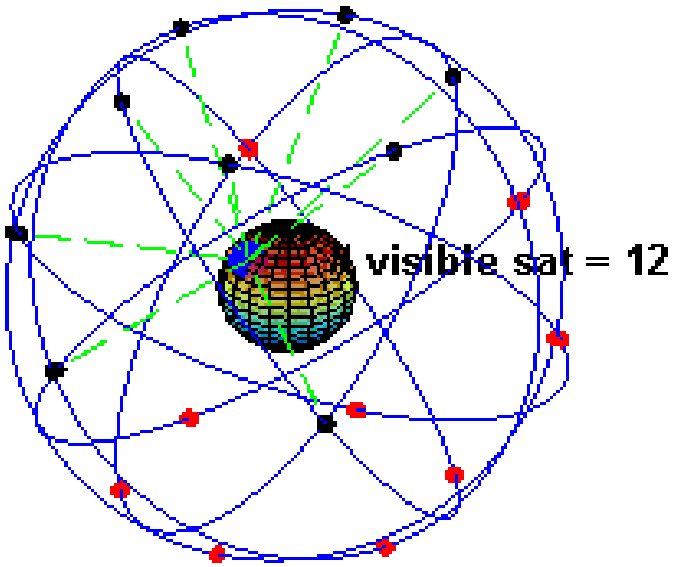



sparkfun ELECTRONICS

# It's GPS time! Grab the software...

- Download from the class GPS directory
- OR go to http://github.com/mgrusin/GPS_class
- Click “download ZIP” (or clone if you already use Git)
- Unzip to a convenient location



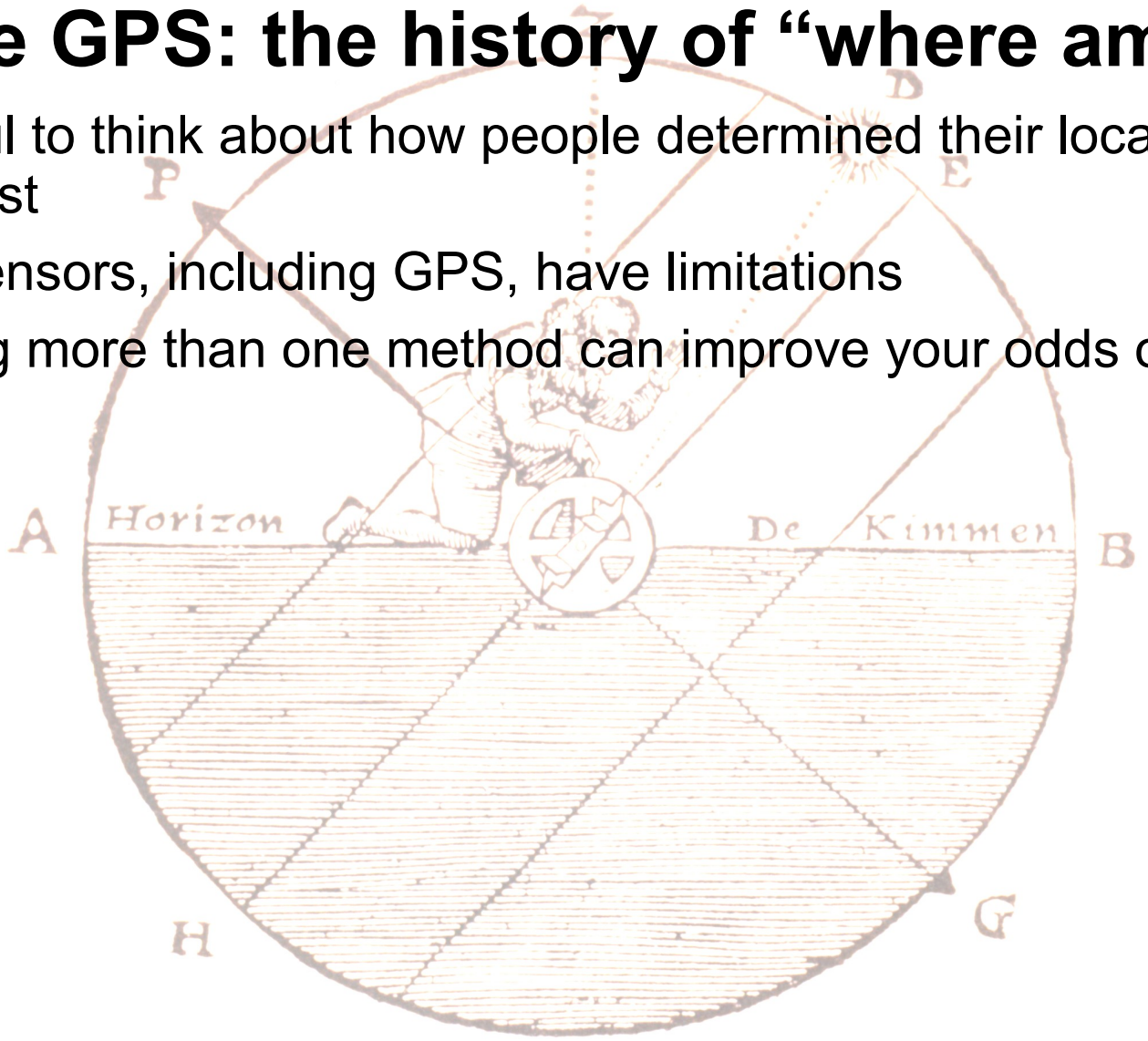

You'll also need this hardware:

- Arduino
- USB cable
- GPS Shield
- EM-506 module
- Serial LCD display
- Battery pack w 4xAA

sparkfun ELECTRONICS

# Before GPS: the history of “where am I?”

- It's useful to think about how people determined their location in the past
  - All sensors, including GPS, have limitations
  - Using more than one method can improve your odds of success!

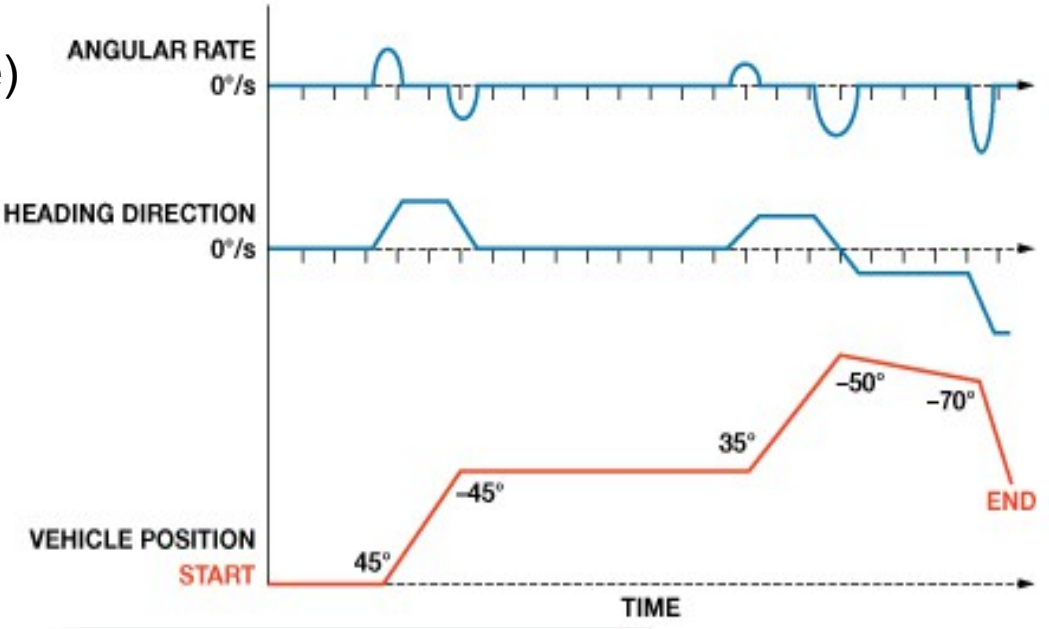


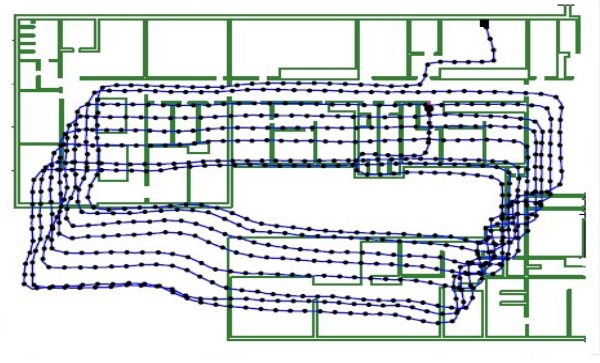

sparkfun ELECTRONICS

# Before GPS: the history of “where am I?”

**“Dead Reckoning”**

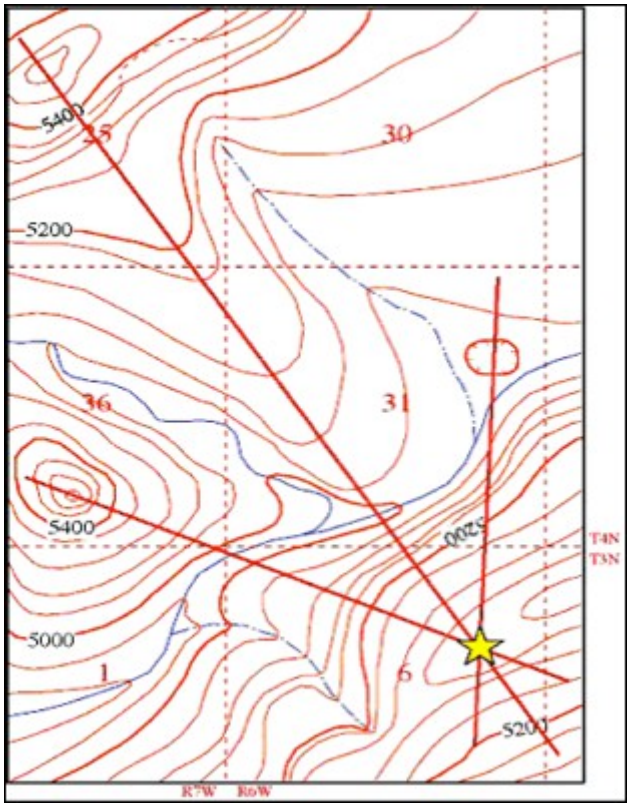
- Keep track of how far you've gone (distance) and the angle you're traveling (azimuth)
  - Distance sensors:
    - Odometry (count wheel rotations)
    - Optical flow detectors (e.g. computer mice)
  - Angle sensors:
    - Potentiometer on “steering wheel”
    - Magnetometer / compass
    - Gyro / IMU (Intertial Measurement Unit)
- Small errors will add up the longer you do this
  - Filtering is important (and complex)



sparkfun ELECTRONICS

# Before GPS: the history of “where am I?”




## Triangulation

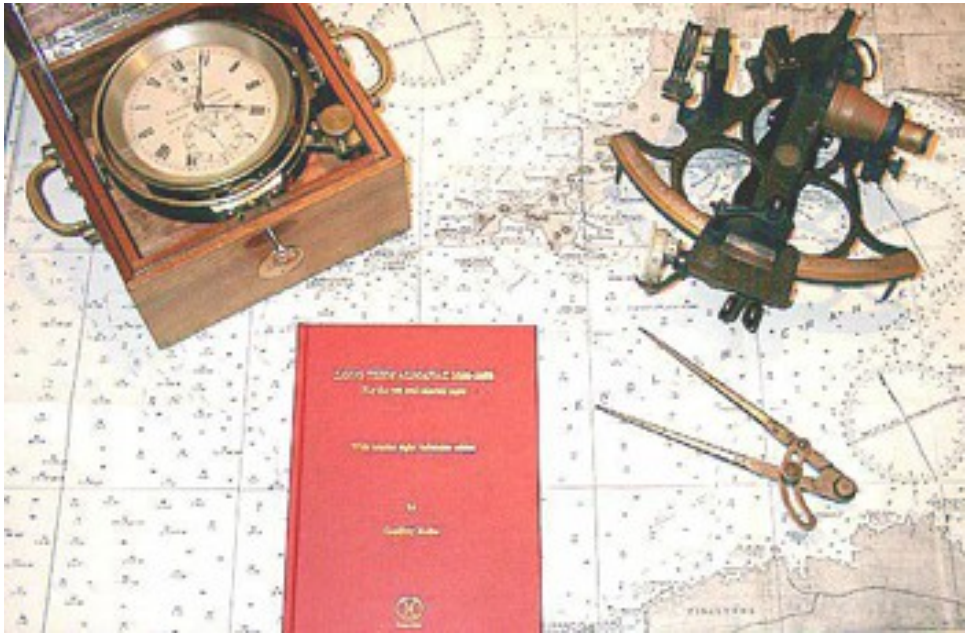
- Measure the bearing angle (azimuth) from your (unknown) position to fixed landmarks with known positions
- Plot reciprocal angle from landmarks
- Your location is where the lines cross!
- Works well, but requires known fixed landmarks and a way to accurately measure bearing angles to them (compass), or the angle between them (protractor or sextant)
- GPS uses a variant of this method!

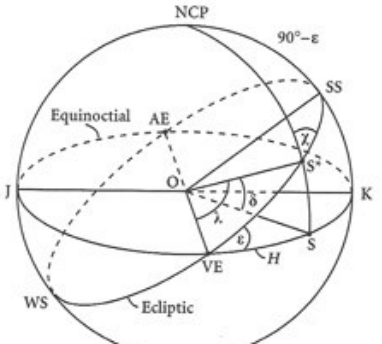
sparkfun ELECTRONICS

# Before GPS: the history of “where am I?”

**Celestial navigation**

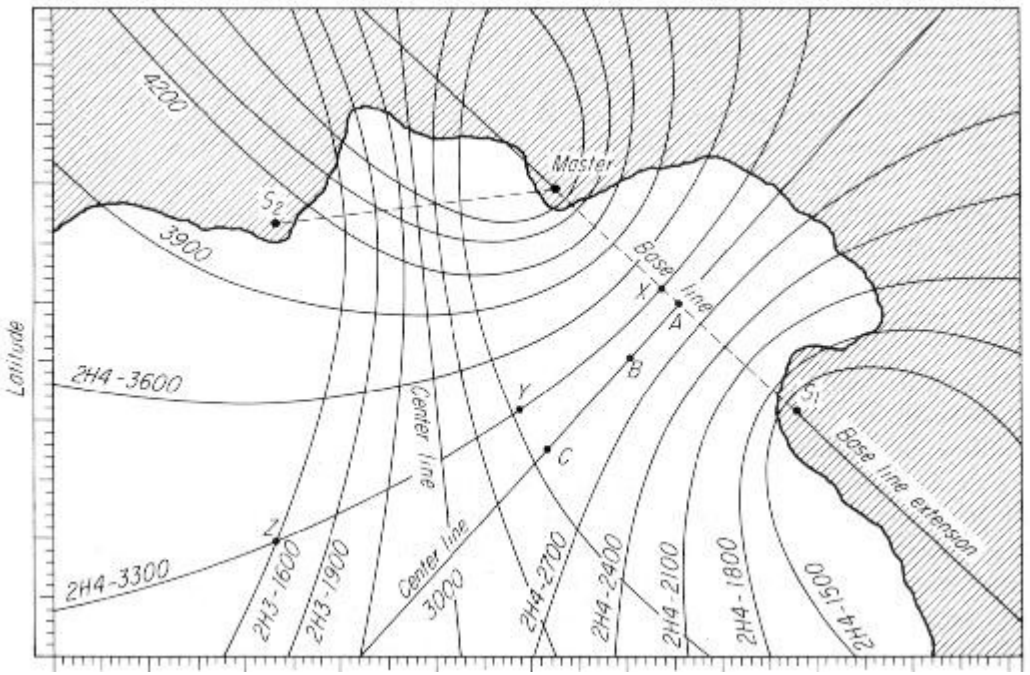
- A variant of triangulation
- A sextant measures precise angle (altitude) between a celestial object and the horizon
- By knowing the exact time and consulting tables, you can determine your location (to ¼ mile if skilled)
- You need the exact time (this was very difficult with mechanical clocks!) and a clear view of the sky
- Still used as a backup (salt water and electronics don't mix)






sparkfun ELECTRONICS

# Before GPS: the history of “where am I?”




## LORAN (Long Range Navigation)

- Developed in the 40s for military and eventually civilian navigation
- Used phase-linked fixed radio transmitters
- The phase difference between two transmitters locates you on a hyperbolic line
- Adding a third transmitter puts you at the intersection of two hyperbolic lines
- Accuracy of 0.1 to 0.25 miles
- Transmitters turned off and system decommissioned in 2010 due to ubiquity, higher accuracy, and lower cost of GPS

sparkfun ELECTRONICS

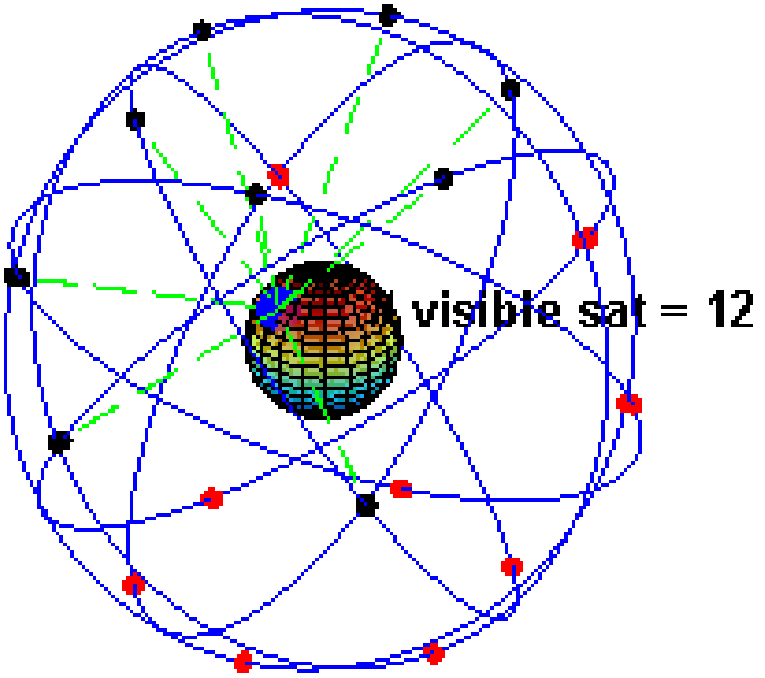
# GPS

- Evolution of earlier military satellite experiments and navigation systems (1960s and 1970s)
- Requires 24 satellites for continuous global operation
  - First launch in 1989, fully operational in 1994
  - Launches continue today for upgrades and maintenance
- Run by USAF 50th Space Wing, based in Colorado Springs
- “One of the most beneficial and humanitarian results of military and aerospace technology”
- Other countries have their own systems: Russia (GLONASS), EU (Galileo), China, India

# How does it work? (the simplified version)

**The GPS constellation**

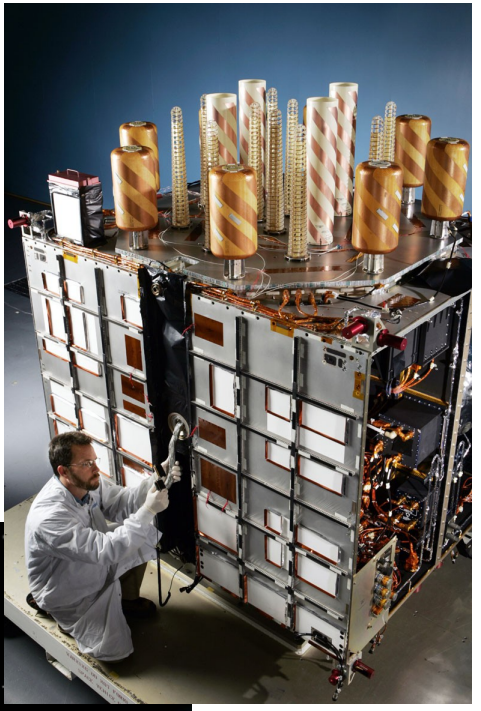
- Consists of at least 24 active satellites (plus spares and upgrades)
- Satellites are in 6 orbital planes (at least 4 active satellites per plane)
- 12500 miles up, 12-hour orbit
- The orbital geometry ensures that (depending on the local terrain) you can see at least four satellites from any point, and often more (8 to 10)




sparkfun ELECTRONICS

# Atomic clocks

- Each GPS satellite contains four(!) Rubidium atomic clocks
  - For redundancy and testing against each other
  - Accurate to better than 1ns (1 billionth of a second)
  - The clocks are designed to tick slightly slow to remove relativistic effects from the satellite's velocity
- Note that even the cheapest GPS receiver gives you access to extremely accurate time!

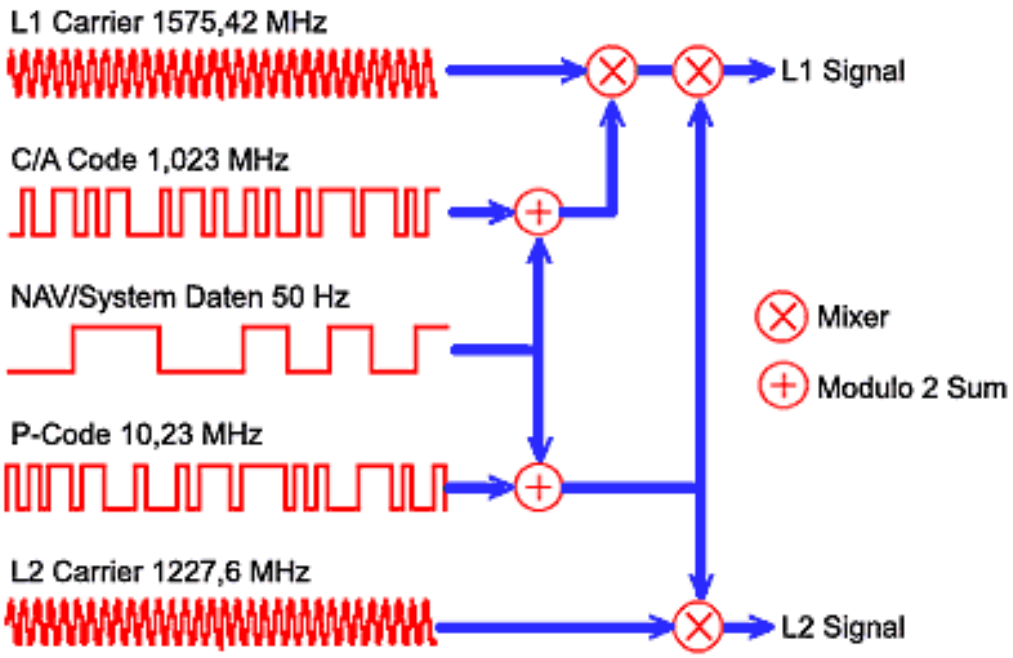


sparkfun ELECTRONICS

# How does it work? (the simplified version)

**Your receiver listens to a cacophony of signals to figure out where it is**

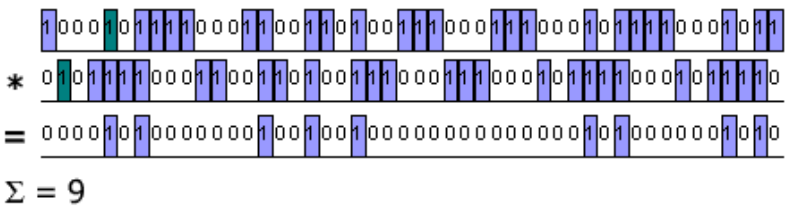
- All active satellites constantly transmit precisely-timed signal patterns
  - Frequencies: 1228MHz and 1575MHz
  - Very low power: 50W
- Data contains fixed sequences unique to each satellite, plus (all) satellite orbits (ephemerides) and the atomic time
- Complete data is 1500 bits, and takes 30 seconds / frame
  - Low rate, but bit timing is extremely precise
- Note that all satellites transmit simultaneously on the same frequencies – everything gets mashed together!

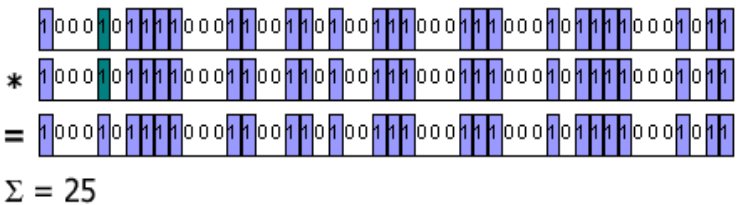



sparkfun ELECTRONICS

# What are “Channels?”

- GPS receivers are advertised as having x “channels” (up to 60!)
- Why so many “channels” with only 24 active satellites?
- These are not receivers (there are only two frequencies, so you would need at most two receivers)
- These “channels” (called “correlators”) do the mathematically-intensive job of trying to match the key patterns (different for each satellite) against the mashed-up signal being received (all the visible satellites mixed together)
- When a channel finds a match, the satellite data can be extracted and used to create or improve the current fix.
- Multiple correlators greatly speed up the initial fix, as they can all work in parallel to identify the satellites in-view.

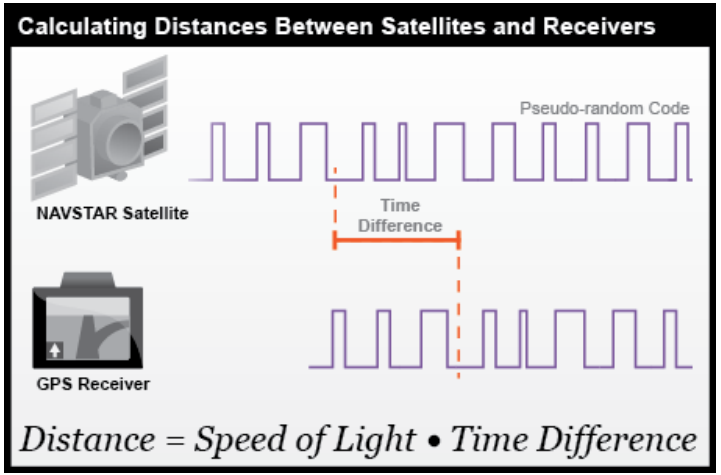


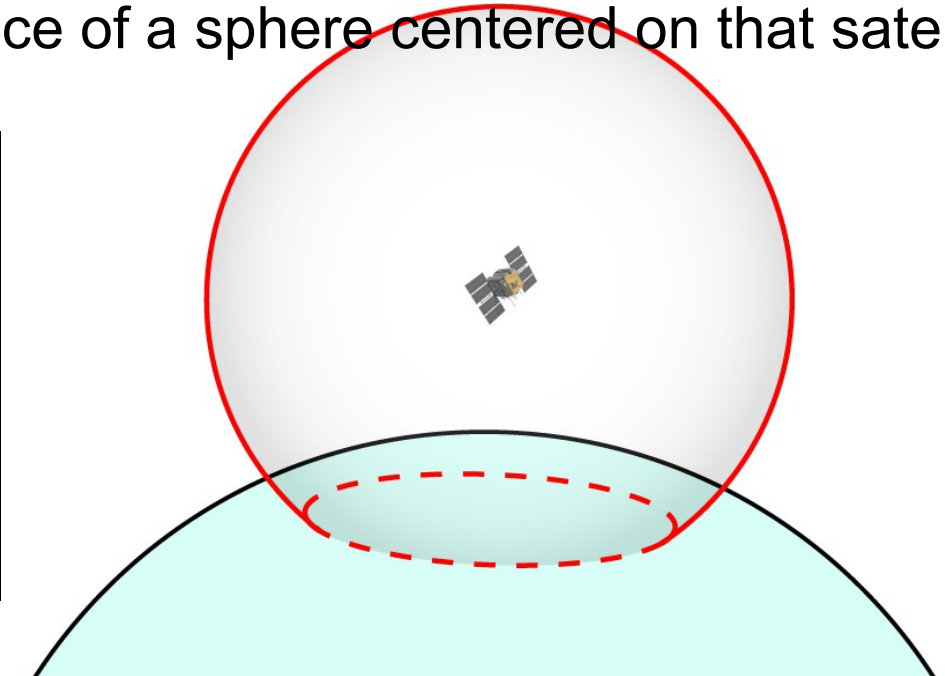




sparkfun ELECTRONICS

# How does it work? (the simplified version)

- When a satellite is found, the atomic clock time on the satellite (sent along with the data) is compared to the (possibly inaccurate) clock time in the GPS receiver, to guess the range (“pseudorange”) to the satellite.
- Calculating the range to a single satellite puts your position somewhere on the surface of a sphere centered on that satellite.

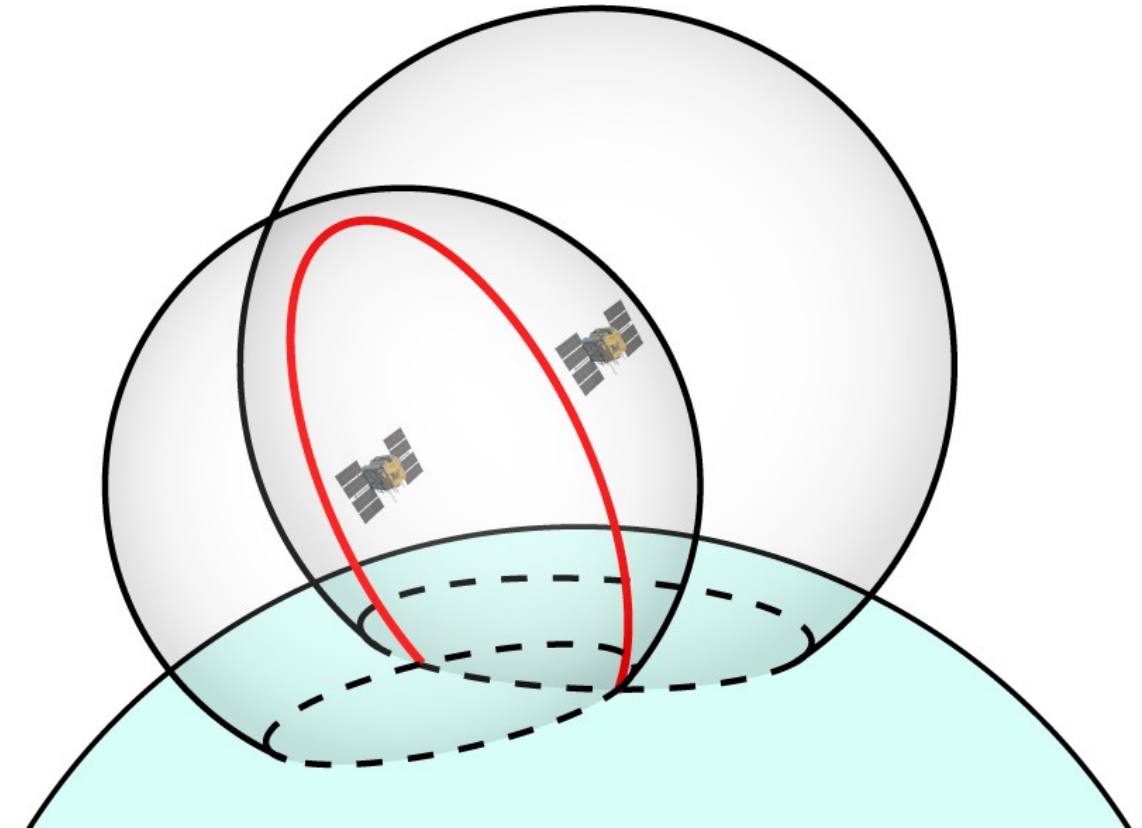


sparkfun ELECTRONICS

# How does it work? (the simplified version)

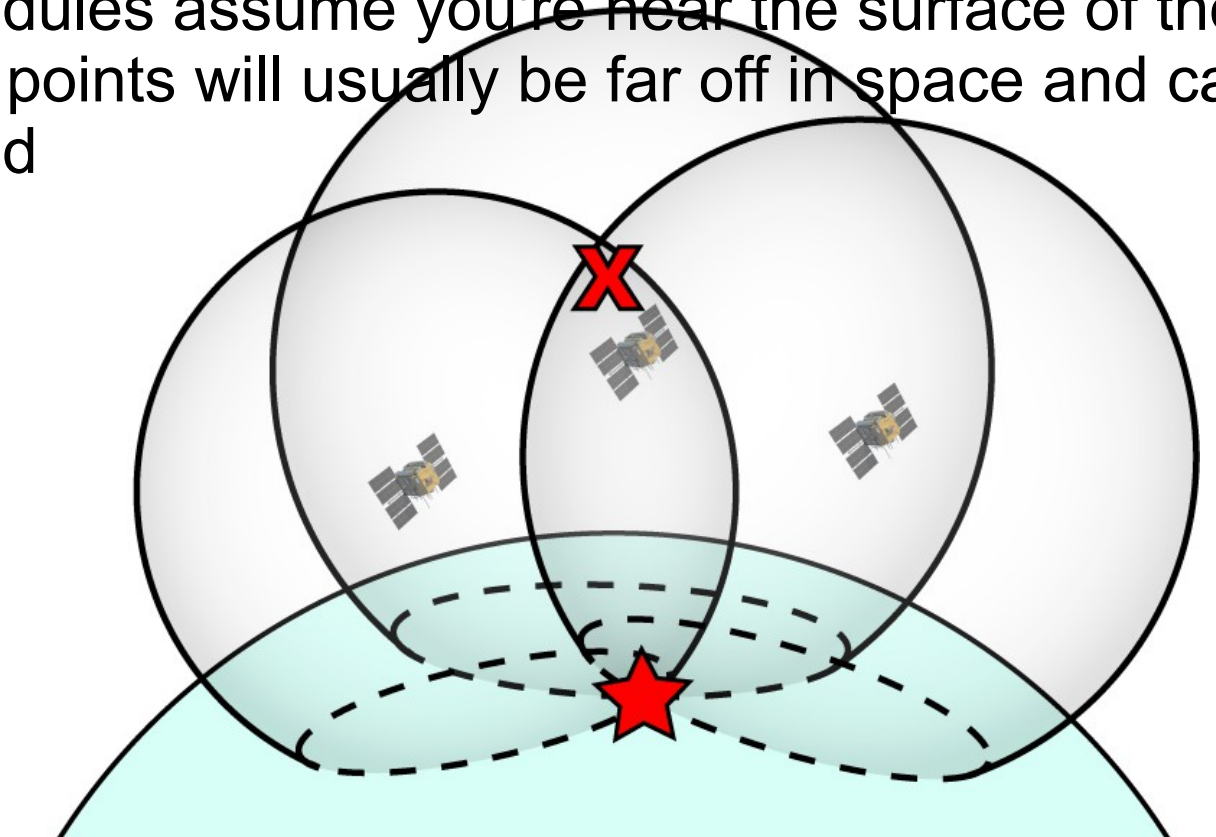
- When you have the range to two satellites, your position will be somewhere on the circle where the two spheres intersect



sparkfun ELECTRONICS

# How does it work? (the simplified version)

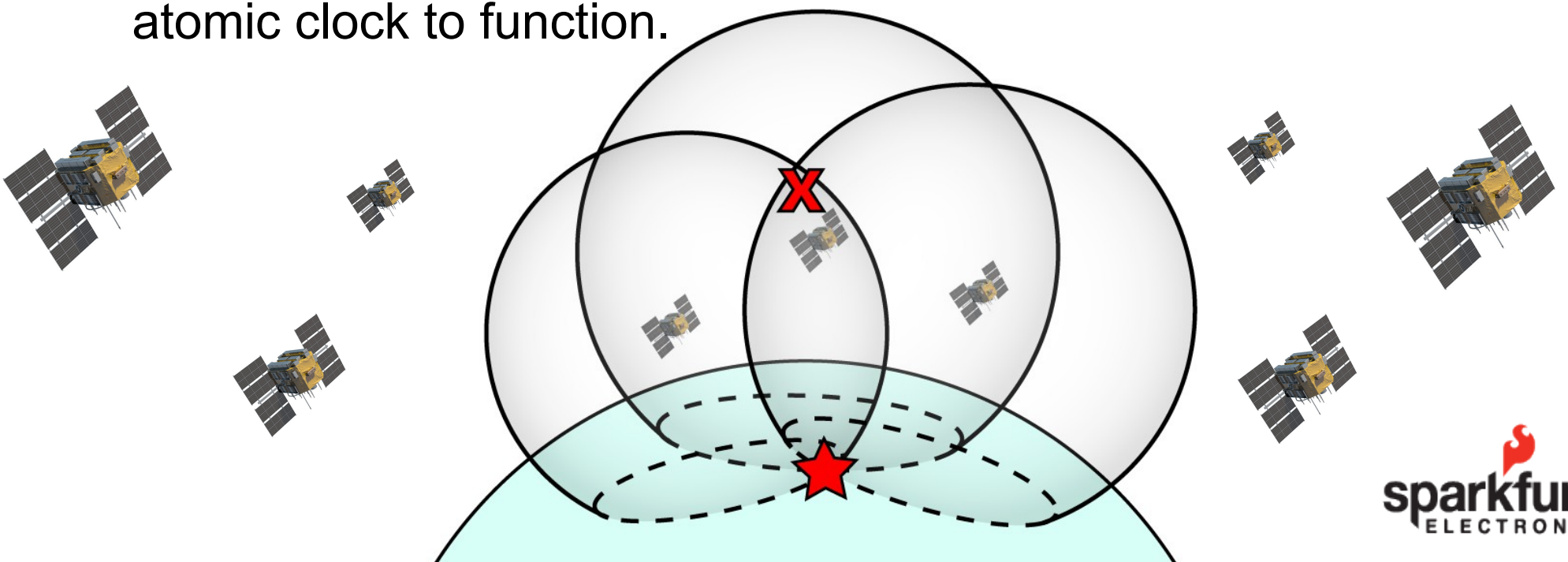
- When you have the range to three satellites, your position will be at one of two points where the three spheres intersect
- GPS modules assume you're near the surface of the earth; one of these points will usually be far off in space and can be discarded




sparkfun ELECTRONICS

# How does it work? (the simplified version)

- When you have four (or more) satellites, you will not only improve the accuracy of the fix, but you can solve for the precise time at the GPS receiver.
- This is why and how the receiver does not need an (expensive!) atomic clock to function.



sparkfun ELECTRONICS

# More history: SA (Selective Availability)

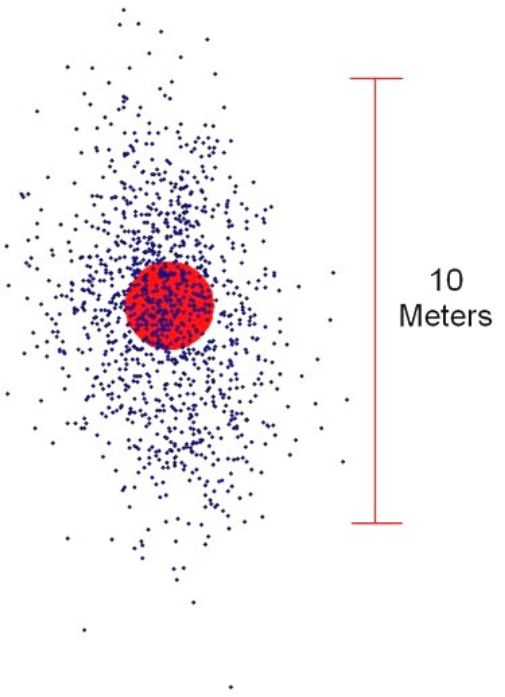
- SA was a military-imposed feature designed into the first-generation of GPS satellites which purposely added random errors to the signal
- Degraded civilian GPS receivers to 100m accuracy
- The military used special receivers and a daily key to remove those errors
- SA was turned off during the entirety of the Gulf War (1990), because troops were buying civilian GPS receivers due to a shortage of military receivers
- Following the war, the US government was increasingly pressured to turn off SA to enhance valuable civilian uses of GPS (especially within the FAA)
- Permanently turned off in 2000
- The military claims to have “methods” to deny GPS to hostile forces (while friendly use is unaffected)
- Newest replacement satellites do not have the capability to add SA errors, so it's not coming back :)

sparkfun ELECTRONICS

# Accuracy and error sources

**How accurate is GPS?**

- When viewing GPS output on a map, you will see the fix wander around the true location
- Error sources:
  - Ionospheric effects ± 5 meters
  - Satellite orbits determination ± 2.5 meter
  - Atomic clock errors ± 2 meter
  - Multipath effects ± 1 meter
  - Tropospheric effects ± 0.5 meter
  - Calculation rounding errors ± 1 meter
- **Worst case: ± 12 meters (~40 feet radius)**
- The GPS receiver can give you metrics (HDOP and CEP)




sparkfun ELECTRONICS

# Improving accuracy

- Use GPS outdoors with a clear view of the sky
  - Signals may or may not penetrate building construction, but will be degraded in all cases
- Avoid trees, rain
  - GPS frequencies are absorbed by water
- Avoid “multipath” situations
  - Urban (or rock) canyons cause signals to bounce, causing phase issues
- Use a good antenna
  - If its directional, keep it pointed up (or it will make things worse!)
- You can also throw money at the problem...

sparkfun ELECTRONICS

# Improving accuracy

You can get specialized GPS receivers that are accurate to several cm, but they are expensive ($1000+). They use tricks to improve on the raw accuracy of GPS

- **DGPS (Differential GPS)**
  - Uses two GPS receivers, one at a known fixed location, and another one (which can be moving) where you want a precise position (AVC robot, etc.)
  - The fixed receiver measures the error (it's fixed, so any difference is error)
  - The fixed receiver transmits the errors to the moving unit. If they're fairly close, they should be seeing roughly the same ionospheric effects
  - Unfortunately this does NOT work by just putting two GPS units near each other (many people have tried). They must always be using the same set of satellites, which isn't guaranteed.
  - The EM-506 on your shield can receive DGPS data on the second serial port, if you have access to a commercial DGPS transmitter to send that data

# Improving accuracy

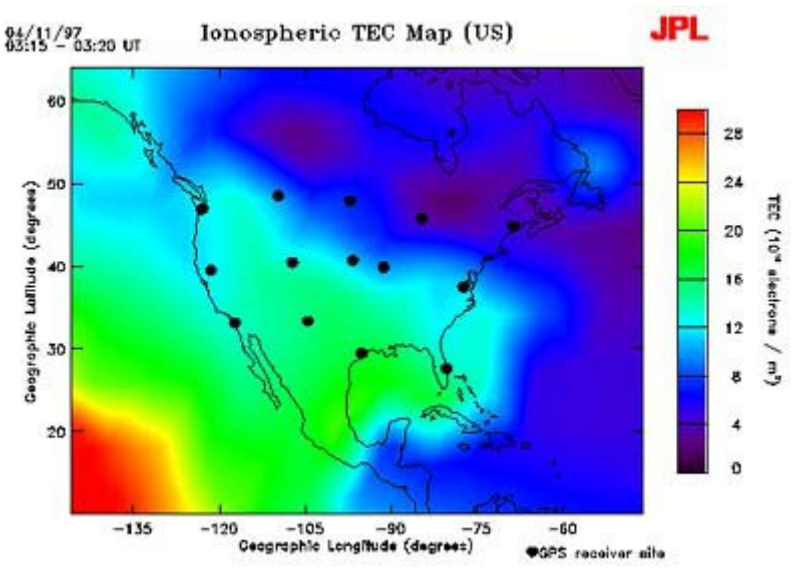
There is also infrastructure to compensate for atmospheric effects, some which cost money and some which are free.

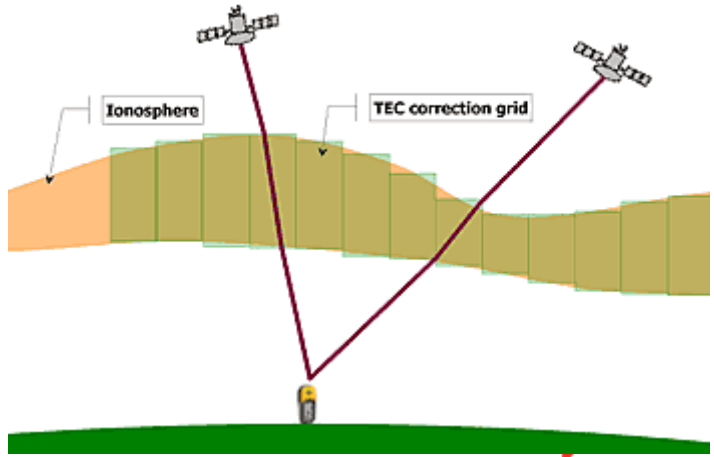
**WAAS (Wide-Area Augmentation System)**

- Less accurate than DGPS, but doesn't require addition hardware

1. FAA measures fluctuations in the ionosphere
2. FAA sends correction data to a geostationary satellite, which broadcasts that data on GPS frequencies as a special “fixed” GPS satellite
3. WAAS-equipped GPS receivers can use this data to correct for ionospheric effects

- Can improve accuracy to ± 2 meters (under ideal conditions)
- **Free to use**, and available on several Sparkfun GPS products including your EM-506




sparkfun ELECTRONICS

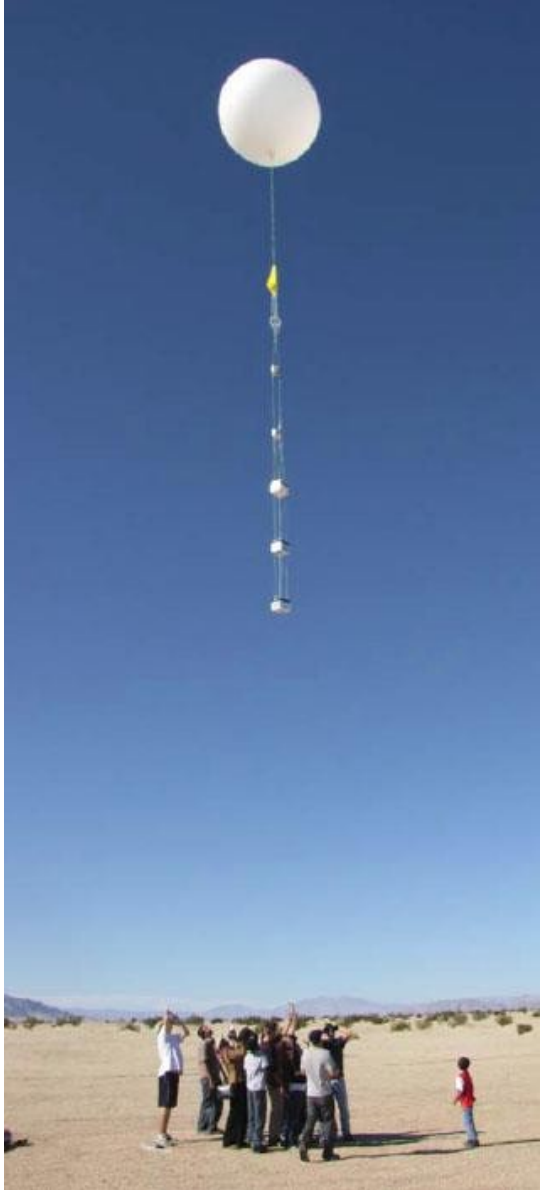
# Improving start-up time

- All GPS receivers need time to get the initial fix (“cold start”)
  - All the correlator channels search for satellites until they find a match
  - Once one satellite is found, it will provide data that speeds up the search
- Under ideal conditions, this will take about a minute, but it could be many minutes (or never) if you're indoors or in another poor-reception area.
- **Many GPS receivers (not the EM-506) allow you to connect a back-up power source to keep the GPS receiver's internal clock running while it's turned off (“warm start”)**
  - This can be an extra connection to your power source, separate battery (coin cell), super-capacitor, etc.
    - **The EM-506 has a large built-in capacitor which works great for 10s of minutes**
    - For other receivers, the GPS Shield has a footprint on the bottom for a coin cell holder
  - Power draw is very low (microamps)
- By leaving the clock running, when the rest of the GPS unit powers up, it can immediately predict which satellites are in view, and quickly look for just those.
- Warm start performance will degrade the longer it's been asleep and the further it's moved from the last known location, but in general this is very effective.

sparkfun ELECTRONICS

# ITAR restrictions

- To prevent their use in weapons, **all** consumer GPS units sold in the US have built-in restrictions to the maximum speed and altitude at which they'll operate
  - No higher than 60,000' altitude
  - No faster than 1000 knots speed (1150 mph)
- **All** consumer GPS modules are programmed to refuse to operate if one or both of those limits is exceeded
- “One or both” becomes important if you want to use GPS on a high-altitude balloon (to 130,000' or more), which only violates *one* of those limits (altitude).
- Different manufacturers will interpret this rule in different ways (and this isn't often in the datasheets). Check Sparkfun user comments and Google for real-world experiences for prospective modules.



# Getting to know your GPS

- You are now the proud owner of a **GPS Shield** and an **EM-506** GPS module
- The EM-506 sends data via standard (asynchronous) serial port (4800 baud)
- The Shield has a switch that routes the EM-506's serial port to either:
  - Pins 0 and 1 (RX and TX, also used to program the Arduino)
  - Pins 2 and 3 (for use with the SoftwareSerial library)
- Your Arduino has only one hardware serial port. SoftwareSerial lets you create additional “soft” serial ports, which can be very useful
- For this class, we'll use SoftwareSerial
  - Set the “UART/DLINE” switch to “DLINE”
  - Remember that serial data will be sent to pins 2 and 3!
- Red LED on EM-506:
  - Off: powered off
  - On (steady): looking for satellites
  - On (blinking): has computed a position fix (>= 4 satellites)

sparkfun ELECTRONICS

# Talking to your GPS

- Almost all GPS receivers output a data fomat specified by (and generally called) **NMEA** (National Marine Electronics Association)
- This is human-readable ASCII (text) data, sent on a standard serial port, usually at 4800bps (you can change the speed)
- Data is sent as comma-separated values in single lines, or “sentences”
- There are about a half-dozen common sentences that have various useful GPS data in them
- Example:

```
$GPRMC,161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598,,*10
```

*What does it mean?*

sparkfun ELECTRONICS

# Talking to your GPS

```
$GPRMC,161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598,,*10
```

| | | |
|---|---|---|
| Message: | $GP | From GPS receiver |
| Sentence: | RMC | "Recommended Minimum" data |
| UTC Time: | 161229.487 | hhmmss.sss |
| Status: | A | A=valid or V=not valid |
| Latitude: | 3723.2475 | ddmm.mmmm |
| N/S: | N | N=north or S=south |
| Longitude: | 12158.3416 | dddmm.mmmm |
| E/W: | W | E=east or W=west |
| Speed: | 0.13 | knots |
| Course: | 309.62 | degrees true |
| Date | 120598 | ddmmyy |
| Mag Var: | NA | degrees E=east or W=west |
| Checksum: | *10 | Lets you check for errors |
| <CR><LF>: | | End of message |

sparkfun ELECTRONICS

# Let's try it!

- Write and upload this sketch
  - Or load “PassThrough”
- Open the serial monitor and set it to 9600 baud
- You should see NMEA data in the serial monitor window

sketch_jun05a §

```
#include <SoftwareSerial.h>

SoftwareSerial gps(2,3); // RX, TX

void setup()
{
  gps.begin(4800);
  Serial.begin(9600);
}

void loop()
{
  char data;

  while (gps.available())
  {
    data = gps.read();
    Serial.print(data);
  }
}
```

rkfun
CTRONICS

# NMEA “sentences”

- You'll see that each line starts with a '$' and a sentence designator “GPXXX”
- Each sentence has a different set of data (some is duplicated)
  - Google “NMEA sentence” to learn what all the fields are
- What you see is the default set of sentences for this GPS module
- You can send commands to the GPS module to turn sentence types on and off, and control the rate at which they appear
  - For example, you can ask it to send “RMC” sentences once per second, and the “satellites in view” sentence once every 10 seconds (or not at all).
  - See the EM-506 datasheet for info

sparkfun ELECTRONICS

# Interpreting live data

**There are any number of GPS viewing programs (free and $) that can read standard NMEA output**

- Download and install the “GPS Viewer / Configuration Software” available from the Venus GPS product page at Sparkfun.com
  - Note that this software is for a different GPS module (Venus) so it can't configure anything, but it will display the NMEA data
- Set the correct COM port and baud rate (9600 for the passthrough)
- Click the “Connect” icon next to the baudrate
- The software will graphically display the position, satellites in view, etc. (but it does not have access to actual geographic data)

sparkfun ELECTRONICS

# Saving / Logging raw data

**If you save the NMEA output of your GPS, you'll have a timestamped record of everything it does**

- Run the previous PassThrough sketch
- Start Coolterm (or the terminal software of your choice) and tell it to talk to your Arduino's serial port at 9600 baud
- Tell Coolterm to save the streaming NMEA data to a text file
- Go somewhere interesting with your Laptop and GPS receiver
- When you're done, tell Coolterm to stop saving the data

**You can now load that data into any software that reads NMEA, e.g. Google Earth:**

- Find a website that translates NMEA sentences into Google Earth .kml files
  - http://www.h-schmidt.net/NMEA/
  - Many others out there, Google “NMEA to kml”
- Give the site your saved NMEA file
- Load the output .kml file into Google Earth!

sparkfun ELECTRONICS

# LIVE data to Google Earth

- Jeff Branson, everybody!

sparkfun ELECTRONICS

# Using GPS data in your sketches

- All the data you need is buried somewhere in one of the NMEA sentences
- You could write a sketch that digs through the data and extracts the values you need...
- But as is common with Arduino, someone has done the hard work for you!
- Use the **TinyGPS** or **TinyGPSPlus** libraries by Mikal Hart
  - We'll use TinyGPSPlus as it's newer and easier to use
- http://arduiniana.org/libraries/tinygpsplus
- All you need to do is:
  1. Blindly pass NMEA sentence data to the library
  2. The library will parse the data and extract the useful information
  3. The library provides easy-to-use functions to give you that data

*Let's try it!*

sparkfun ELECTRONICS

# Install TinyGPSPlus

1. Go to http://arduiniana.org/libraries/tinygpsplus/
2. Click download, choose the latest, click “source code (zip)”, save.
3. Extract the zip file and view the contents
4. Rename e.g. “TinyGPSPlus-0.94b” to “TinyGPSPlus”

   Arduino doesn't like punctuation except for “_” in filenames
5. Drag “TinyGPSPlus” to your “Arduino/libraries” folder
6. Restart the Arduino IDE

sparkfun ELECTRONICS

# Let's try it out!

- In the Arduino IDE, open:
  File/
  Examples/
  TinyGPSPlus/
  DeviceExample
- Change this line:

```
static const int RXPin = 4, TXPin = 3;
```

  to match the GPS Shield connections:

```
static const int RXPin = 2, TXPin = 3;
```

- Upload the sketch, and open the serial monitor to 115200 baud
- You can run any of the examples as long as you change the above line in each sketch

sparkfun ELECTRONICS

# Treasure Hunt

- The TinyGPSPlus library has functions that can tell you how far away and what direction you are from another location.
- Let's use that to make a portable treasure hunting device, and go outside and find some treasure!

sparkfun ELECTRONICS

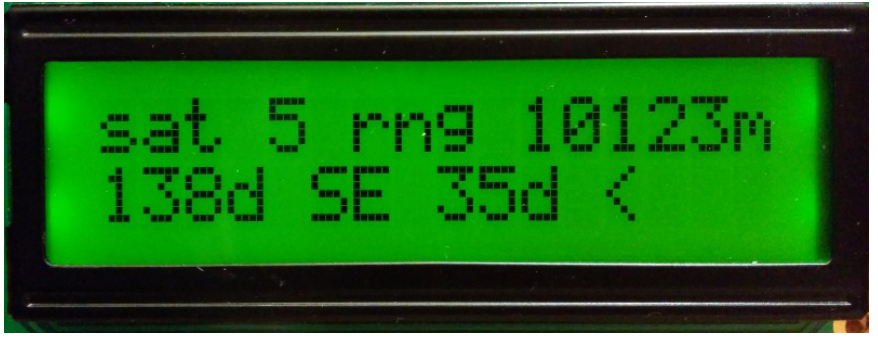
# Treasure Hunt

1. Connect your serial LCD to your Arduino + GPS Shield
    1. Black wire to GND
    2. Red wire to 5V
    3. Yellow wire to pin 5
2. Load the TreasureHunt sketch
3. Take one *Treasure Coordinate Slip* from the HAT OF DESTINY
4. Edit the sketch to replace TARGET_LAT and TARGET_LON (defined at the top of the sketch) with your treasure coordinates (be sure to double check the numbers for accuracy!)
5. Upload the sketch...

sparkfun ELECTRONICS

# Treasure Hunt

- The TreasureHunt sketch opens two SoftwareSerial ports, one for the GPS (pins 2 and 3) and one for the LCD (pins 4 and 5)
- It uses the TinyGPSPlus library to compute the distance and bearing from the GPS location to your treasure location, and display useful data on the LCD. Once you have a GPS lock (flashing red light on the EM-506), you should see something like this:

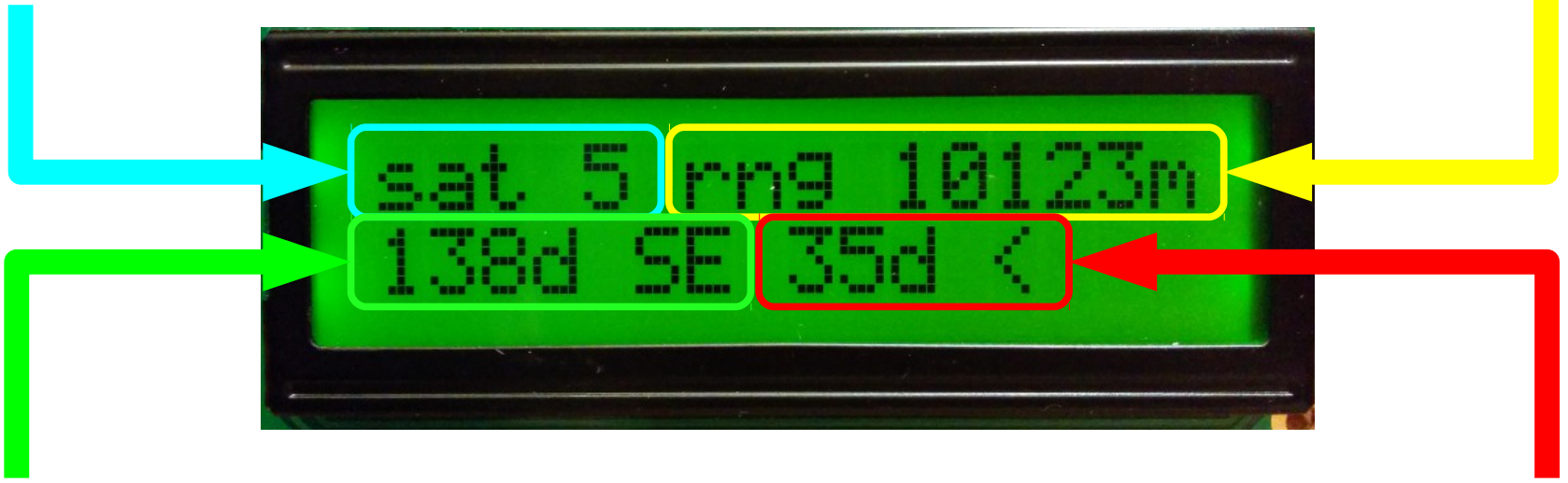



sparkfun ELECTRONICS

# Treasure Hunt

Number of satellites the GPS receiver can see (the more the better)

Range in meters from the current position to the target position



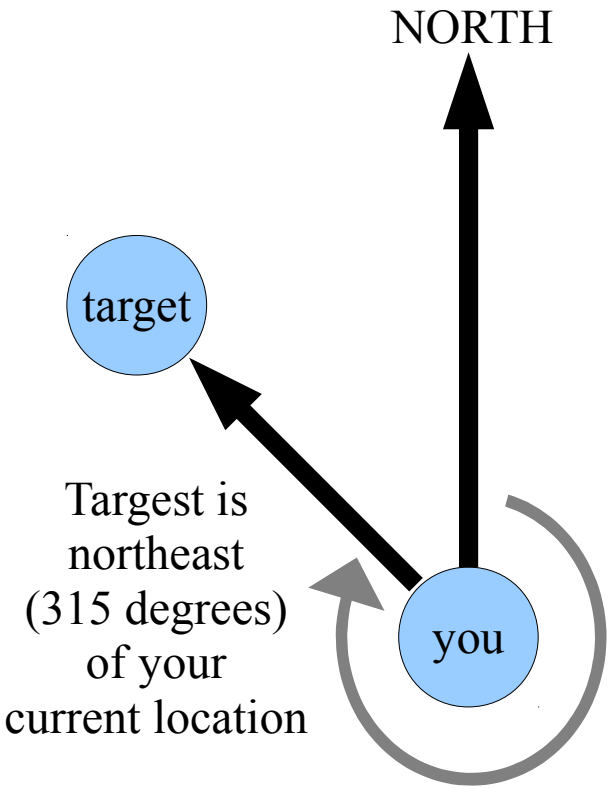

Absolute (compass) bearing from the current position to the target

*Relative* bearing from your current **direction** to the target (must be moving to work!)

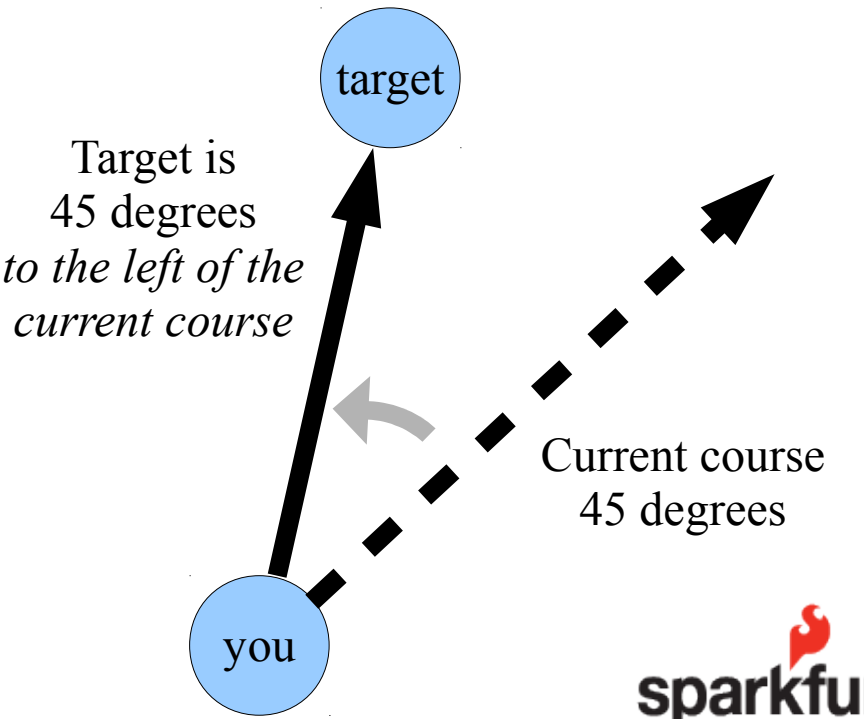
sparkfun ELECTRONICS

# Absolute vs. Relative course

Absolute course
(no movement required)




Relative course
(requires movement!)




sparkfun ELECTRONICS

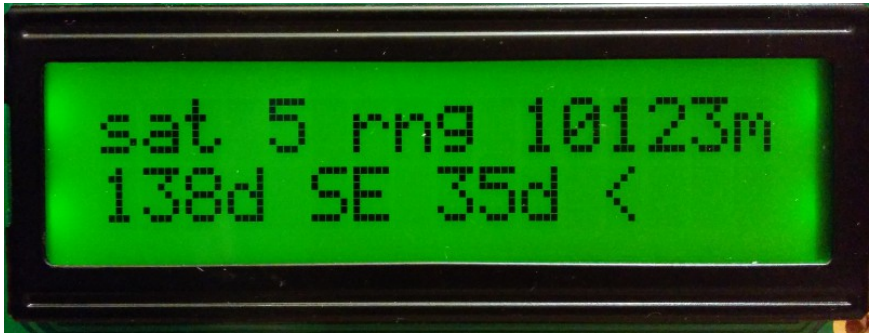
# GPS course determination

**GPS units can determine your course and speed, *but these numbers are only valid if you're moving***

- Course and speed are determined by looking at the difference between the current position and several previous positions.
- If you're moving, this works well. But if you're not moving, the natural meandering of the GPS location error will give you “windmilling” course and speed information.
- If you stop, you won't know what direction you're facing, which can be problematic for robots
- Some GPS units include an actual magnetic compass to provide heading information even if you're standing still
  - You can also incorporate a separate compass into your projects
- Some GPS units have a “pinning” function that will freeze the location if no movement is suspected

# Treasure Hunt

- Make sure your GPS LED is blinking (has a position fix) and information is displayed on the LCD
- If everything is working, disconnect the USB cable, and connect your 4 x AA battery pack
- If everything is STILL working, go outside and find your treasure!




sparkfun ELECTRONICS

# Thanks for spending time with us!

- More information
  - http://learn.sparkfun.com
  - http://www.kowoma.de/en/gps/
  - mike.grusin@sparkfun.com

# Treasure hunt!

| | | |
|---|---|---|
| 1. | 40.064367, -105.210019 | main sparkfun sign |
| 2. | 40.064594, -105.209897 | east front sign |
| 3. | 40.064596, -105.210328 | west parking sign |
| 4. | 40.065326, -105.210346 | back parking light pole |
| 5. | 40.065324, -105.209903 | back parking light pole |
| 6. | 40.065132, -105.209670 | back parking light pole |
| 7. | 40.064975, -105.209425 | back parking light pole |
| 8. | 40.064908, -105.209527 | weird pond pole |
| 9. | 40.065082, -105.210384 | tree at corner of building |
| 10. | 40.065090, -105.209854 | tree at corner of building |
| 11. | 40.064664, -105.209868 | tree at corner of building |
| 12. | 40.064669, -105.210390 | tree at corner of building |

sparkfun ELECTRONICS