Ultimate GPS
GPS “segments”
**Frequency Bands**

**GPS L1 Band:** 1575.42 MHz with a bandwidth of 15.345 MHz  
**GPS L2 Band:** 1227.6 MHz with a bandwidth of 11 MHz  
**GPS L5 Band:** 1176.45 MHz with a bandwidth of 12.5 MHz
Trilateration NOT Triangulation
Trilateration NOT Triangulation
GPS Basics

- -165 dBm sensitivity, 10 Hz updates, 66 channels
- 5V friendly design and only 20mA current draw
- Breadboard friendly + two mounting holes
- RTC battery-compatible
- Built-in datalogging
- PPS output on fix
- Internal patch antenna + u.FL connector for external active antenna
- Fix status LED
Specs

**Technical Details:**
- Satellites: 22 tracking, 66 searching
- Patch Antenna Size: 15mm x 15mm x 4mm
- Update rate: 1 to 10 Hz
- Position Accuracy: 3 meters (all GPS technology has about 3m accuracy)
- Velocity Accuracy: 0.1 meters/s
- Warm/cold start: 34 seconds
- Acquisition sensitivity: -145 dBm
- Tracking sensitivity: -165 dBm
- Maximum Velocity: 515m/s
- Vin range: 3.0-5.5VDC
- MTK3339 Operating current: 25mA tracking, 20 mA current draw during navigation
- Output: NMEA 0183, 9600 baud default, 3V logic level out, 5V-safe input
- DGPS/WAAS/EGNOS supported
- FCC E911 compliance and AGPS support (Offline mode : EPO valid up to 14 days)
- Up to 210 PRN channels
- Jammer detection and reduction
- Multi-path detection and compensation
GPS Code Breakdown

```
58 #include <Adafruit_GPS.h>
59 #if ARDUINO >= 100
60 #include <SoftwareSerial.h>
```

GPS is imported like a library

Serial uses the RX and TX pins

```
77 // If using software serial, keep these lines enabled
78 // (you can change the pin numbers to match your wiring):
79 // #if ARDUINO >= 100
80 // SoftwareSerial mySerial(3, 2);
81 // #else
82 // NewSoftSerial mySerial(3, 2);
83 // endif
84 // Adafruit_GPS GPS(&mySerial);
85 // If using hardware serial (e.g. Arduino Mega), comment
86 // out the above six lines and enable this line instead:
87 Adafruit_GPS GPS(&Serial2);
```
/ 9600 NMEA is the default baud rate for MTK - some use 4800
GPS.begin(9600);

// You can adjust which sentences to have the module emit, below

// uncomment this line to turn on RMC (recommended minimum) and GGA (fix data) including altitude
//GPS.sendCommand(FMTK_SET_NMEA_OUTPUT_RMCGGA);

// uncomment this line to turn on only the "minimum recommended" data for high update rates!
//GPS.sendCommand(FMTK_SET_NMEA_OUTPUT_RMCONLY);

// uncomment this line to turn on all the available data - for 9600 baud you'll want 1 Hz rate
//GPS.sendCommand(FMTK_SET_NMEA_OUTPUT_ALLDATA);
GPS Update Rate

// Set the update rate
// Note you must send both commands below to change both the output rate (how often the position
// is written to the serial line), and the position fix rate.
// 1 Hz update rate
GPS.sendCommand(PMTK_SET_NMEA_UPDATE_1HZ);
GPS.sendCommand(PMTK_API_SET_FIX_CTL_1HZ);
// 5 Hz update rate - for 9600 baud you'll have to set the output to RMC or RMCGGA only (see above)
GPS.sendCommand(PMTK_SET_NMEA_UPDATE_5HZ);
GPS.sendCommand(PMTK_API_SET_FIX_CTL_5HZ);
// 10 Hz update rate - for 9600 baud you'll have to set the output to RMC only (see above)
// Note the position can only be updated at most 5 times a second so it will lag behind serial output.
GPS.sendCommand(PMTK_SET_NMEA_UPDATE_10HZ);
GPS.sendCommand(PMTK_API_SET_FIX_CTL_5HZ);
// Interrupt is called once a millisecond, looks for any new GPS data, and stores it
SIGNAL(TIMER0_COMPA_vect) {
  char c = GPS.read();
  // if you want to debug, this is a good time to do it!
  if (GPSECHO)
    if (c) UDR0 = c;
  // writing direct to UDR0 is much much faster than Serial.print
  // but only one character can be written at a time.
GPRMC Data: GMT (HHMMSS.ms), Status Code, Geolocation Data (N/E = +, S/W = - for gmaps)

Latitude: DDMM.MMMM and Longitude: DDDMM.MMMM

```$GPRMC,185240.677,v,,,,0.00,0.00,280918,,,,N*4B
$PGTOP,11,2*6E
$GPRMC,185241.677,v,,,,0.00,0.00,280918,,,,N*4A
```

**Latitude = 4211.25**

**Longitude = 8822.06**
Data Fix

200 if (GPS.newNMEAreceived()) {
201     GPS.parse(GPS.lastNMEA());
202 }

249 // Using GPS.latitude and GPS.longitude do not return useful data
250     Serial.print("Latitude = "); Serial.print(GPS.latitude);
251     Serial.print("\nLongitude = "); Serial.print(GPS.longitude);
252     myFile.print("\n");
253     myFile.print("\nLatitude = ");
254     myFile.println(GPS.latitude);
255     myFile.print("\n");
256     myFile.print("Longitude = ");
257     myFile.println(GPS.longitude);