
ET301 GPS-UAV Development Platform

Part 1: Hardware description

ET301 GPS-UAV Development Platform

Background

When Maynard Hill's autonomous RC plane made its historic flight across the Atlantic Ocean a few years ago, Brian Premerlani said to his dad, "Why don't we do something along those lines? It might be fun. How hard can it be?" Indeed. Brian and his dad recognized that getting an RC model airplane to fly across the Atlantic was more than they wanted to commit to, so they set their sights a little lower. They settled on getting an RC sailplane to fly by itself using readily available components, most of which were purchased from Spark Fun.

To get their feet wet, they started with an even easier goal, getting an RC truck to navigate by GPS. They installed a GPS receiver/antenna and an electronic compass on a rapid-prototyping board and after some modest development and debugging effort, were successful with the truck.

They then moved up to an RC sailplane, starting with rudder control, only. They built their own board with GPS and compass, only to be harshly reminded that a 2-axis compass works properly only when it is level. Switching to a gyro plus GPS solved that problem. Finally, they added a second gyro and accelerometers for pitch control. The firmware that they developed performs the following functions to make flying the sailplane more interesting and easy:

- Manual control. This mode is used for take-off and landing. Signals from the radio are passed straight through to the servos. This mode is also used during debugging when any of the other modes become unstable.
- Augmented manual control. The control interprets signals from the radio as desired turning rate and as angle of attack, and uses feedback from the gyros and accelerometers to stabilize the flight. This mode completely eliminates any tendency for the sailplane to "porpoise"¹ and steadies the turns against buffeting winds. Stability is improved so that it is possible to fly more slowly than the normal stall speed of the sailplane. Rudder-elevator mixing is performed automatically so that it is not necessary to use "up-elevator" during turns.²
- Circling in thermals. Using the radio, the pilot signals the controller to lock-in the present location. The sailplane then autonomously circles that location until it receives a command to resume augmented control or manual control.

¹ "Porpoising" is a term familiar to RC sailplane pilots. When a lightly loaded sailplane is flown with too much up elevator, it climbs, loses speed, and eventually stalls. When it stalls, it pitches forward into a dive, picks up speed, and pitches back up into a climb. The cycle repeats. The trajectory of the plane resembles that of a porpoise swimming and jumping out of the water.

² Pilots of RC sailplanes know that you have to use some up-elevator in turns to compensate for the tendency for the plane to spiral downward when turning.

- Safe return. If the transmitter is shut off, or if the radio loses the signal, the controller flies the plane back toward the point from which it was launched. After it passes over the point, it then circles the point.

The board they developed, without firmware, is now available from Spark Fun, for those who want to develop their own UAV. It is a 2-axis controller (pitch and yaw) with gyros, accelerometers, GPS, and CPU, suitable for control of high-dihedral³ RC planes and sailplanes. You can use it to develop and implement your own UAV ideas.

This document is the first part in a three part set, covering the GPS-UAV hardware. The second part will review flight dynamics, and control principles. The third part will contain firmware examples.

Hardware

The GPS-UAV is a hardware platform for you to develop your own control and navigation algorithms. *It does not contain any firmware.* It does contain the following hardware resources:

Handling precautions

The GPS-UAV is rather robust, but there are a couple of handling precautions to be aware of:

- A trace on one of the development prototype boards was broken due to excessive twisting during installation of the board in a sailplane, while it was being crammed through a tight spot. The board will tolerate normal handling, but do not bend or twist it on purpose. While you are mounting the board in your plane, if there is something in the way, temporarily remove it.
- The prototype had a small cable tie gathering the entire set of connection cables into a bundle to reduce the amount of flexing where the cables were soldered onto the board. The tie was deliberately left off from the production boards, on the assumption that you might or might not want it, depending on how much handling of the board you anticipate, you how you planned to route the cables. You may want to tie the cables together.

ET301 SiRF III GPS receiver

The heart of the GPS-UAV is the GPS receiver. The ET301 SiRF III is more than adequate for determining location and direction of motion. Complete specifications are available either on the manufacturer's website or at Spark Fun. The ET301 is one of the most sensitive GPS receivers available, and starts up very quickly, even without the optional backup battery installed.

Once per second the GPS reports information useful for navigation including longitude, latitude, altitude, and direction and speed over ground. The direction and speed information from the GPS is much more accurate than what you could compute by subtracting successive pairs of positions, since it is based on a Doppler shift calculation using the baseband signal.

The location information is generally more accurate than advertised by the manufacturer. Furthermore, if you use the GPS-UAV to simply fly around your flying field, you can

³ The term "high-dihedral" is familiar to RC sailplane pilots. The dihedral angle is the angle between the wings, looking along the fuselage. For a sailplane, this angle is fairly large, giving the wings a "V" shape, and giving the sailplane a tendency to fly with its wings level.

improve the accuracy somewhat by navigating in terms of motion relative to a GPS measured reference point, such as the location of your plane when you turn it on.

Communications with the GPS is either with a NMEA standard format or a binary format. The NMEA standard format is recommended because it is easy to use and debug. The only information that you might want to use that is available only through the binary interface is vertical speed. The firmware used to test the prototype did not use vertical speed.

The communications link between the GPS receiver and the CPU is via TX and RX pins of the USART:

From GPS to CPU	RX
From CPU to GPS	TX

The default baud rate of the board is 4,800 and the default protocol is NMEA, which is quite readable. It is suggested that you purchase Spark Fun's ET-301 evaluation board to see what the NMEA protocol is like, and to better understand the behavior of the ET-301.

The GPS-UAV has a battery holder for an optional backup battery for the GPS. The board will work fine without the backup battery, but if you want to reduce the starting time for the GPS, it is recommended that you use a CR1225 backup battery. You probably will want to take the battery out of the board when you are not either doing development or testing: the battery drain is small, but so is the battery. If you leave the battery in continuously, it will run down in a few weeks.

The GPS LED will flash when the GPS is ready.

Information about the ET301 is available on the Spark Fun website, www.sparkfun.com, or on the manufacturer's website, www.usglobalsat.com.

Antenna

There is an SMA connector for an antenna. The Spark Fun website has recommended active antennas listed. There are two antennas that you will probably want to get, a light-weight antenna with a short lead, and a magnetic mount antenna with a very long lead. The magnetic mount is very useful for debugging purposes, the light-weight one is for flight. The magnetic mount antenna has two features that are useful for debugging. The magnetic mount is useful for taking the board on test drives in a car with the antenna attached to the roof. Also, the long wire makes it useful for debugging while indoors with a weaker signal by putting the antenna at a window.

Be careful in handling the flight antenna, the connection to antenna is fragile, you might want to reinforce it with a bit of epoxy.

Proper operation of the GPS does depend broadly on the orientation of the antenna. For best results, orient the plane of the antenna parallel to the ground. Other than that, the GPS does not depend on the orientation of the antenna. As the antenna moves through space the GPS indicates where it is and its vector velocity.

PIC18F2520

The CPU is a Microchip PIC18F2520. Manuals are available from Microchip at www.microchip.com. The PIC18F2520 is ideally suited for use in the GPS-UAV, with plenty of processing power and features. The 18F2520 is a slightly enhanced version of the 18F252, which was used in the development of the first prototype.

4 MHz oscillator

The GPS-UAV has a 4 MHz oscillator that is a legacy of the crystal that came with a Microchip prototyping board that was used to develop a navigator for an RC truck. It turns out that a 4 MHz oscillator and using the 4X frequency phase-locked frequency multiplier option on the PIC produces an on-chip clocking rate of 16 MHz, which is a very convenient frequency for the PIC PWM to produce the right pulse width for control of RC servos. In order to use the frequency multiplier option, you will need to set the corresponding configuration bit. Configuration bits are set through menus on the programmer or debugger interface.

Gyros and accelerometers

There are 2 gyros and a dual axis accelerometer on the GPS-UAV board. For the pitch axis, there is an IMU daughter board with an Analog Devices ADXRS401 gyro and an Analog Devices ADXL203 two axis accelerometer. For the yaw axis, there is another ADXRS401 gyro mounted on the board. Sensitivity of the gyros is approximately 15 millivolts per degree per second, with a maximum range of plus or minus 75 degrees/second. Sensitivity of the accelerometers is approximately 1000 millivolts per g, with a maximum range of plus or minus 1.7 g. Specifications for the gyros and accelerometers are available either on the Analog Devices website, www.analog.com, or on the Spark Fun website, www.sparkfun.com.

The gyros and accelerometers used in the GPS-UAV have rather high gains. That was a deliberate choice in order to improve overall noise and drift performance. The dynamic range is somewhat limited. For most applications that is not an issue. For sailplane applications, the plane will be making gentle turns under UAV control, so the range will not be exceeded. For other applications, it may be desired to exceed the range of the gyros and accelerometers. That is acceptable, because there are techniques for compensating for saturation.

The dual axis pitch accelerometer can be used to measure accelerations parallel to the fuselage and perpendicular to the wings. The pitch gyro measures the rate of rotation around the pitch axis of the plane, the rate at which the nose of the plane is rotating up or down. The yaw gyro measures the rate of rotation around the yaw axis of the plane, the rate at which the nose of the plane is rotating left or right. Keep in mind that the gyros measure rotation rates in the reference frame of the plane, not the reference frame of the ground.

Also note that it is not precisely true that the accelerometers measure acceleration. What each accelerometer actually measures is net acceleration minus gravity, so when the plane is not accelerating, each accelerometer measures the component of gravity along its measurement axis.

The outputs of the gyros and accelerometers are analog voltages. Zero rotation and zero acceleration correspond to half voltage. Voltage referencing for the accelerometers is slightly different than for the gyros. The output voltage of the accelerometer is proportional to the supply voltage. It is recommended that you also use the supply voltage as the reference for the A/D converter in the CPU. In that case, there is no need to compensate accelerometer readings for variations in supply voltage as the batteries discharge.

In the case of the gyros, there is a reference voltage that is generated internally by the gyro, so that the output of the gyro does not vary with supply voltage. In that case, it is recommended to compensate the gyro readings for variations in supply voltage as the batteries discharge. This can be done by reading the gyro reference voltage.

The gyros can drift slightly with temperature. Depending on the details of your firmware and how careful you are about temperature, you probably will not need to worry about

temperature drift. However, if you want to compensate for temperature effects, the gyros have a built-in temperature measurement.

The analog signals from the gyros and accelerometers are read on the following A/D channels of the 18F2520:

Yaw rate	AN0
2.5 volt reference	AN1
Pitch rate	AN2
Acceleration Y	AN3
Acceleration X	AN4
Temperature	AN8

The X and Y acceleration axes and rotation directions are labeled on the gyro/accelerometer daughter boards.

The gyros and accelerometers also have a self test feature. The accelerometer has a self test TTL input pin, labeled ST3 on the daughter board which causes the outputs to shift by about 0.8 volts. The gyros have two self test TTL input pins, one, labeled ST1 on the daughter board which causes an output shift by about -0.8 volts, the other one labeled ST2 creates an output shift of +0.8 volts. The ST3 pin and the two ST1 pins are tied together and connected to an output of the 18F2520. The two ST2 pins are tied together and connected to another output of the 18F2520. Pin assignments are as follows:

ST1/ST3	RC3
ST2	RB3

Self test is activated by applying a logic level 1 to the test pin.

There are some restrictions in mounting the board in your plane for the gyros and accelerometers to work properly. In order for the yaw gyro to work, the plane of the board should be perpendicular to the yaw axis of the plane. In order for the pitch gyro and accelerometers to also work, the long axis of the board must be parallel with the fuselage. There are four orientations that will work:

1. Board flat on the bottom of the fuselage, GPS receiver towards the nose.
2. Board flat on the bottom of the fuselage, GPS receiver towards the tail.
3. Board upside down, flat part facing up, GPS receiver towards the nose.
4. Board upside down, flat part facing up, GPS receiver towards the tail.

Orientations 1 and 2 were used equally well during development of the GPS-UAV. In principle, options 3 and 4 would also work. The signs of the gains of the gyros and accelerometers depend on the orientation you select. Positive directions are indicated on the daughter boards.

Filtering and sampling rate

The accelerometers and gyros have built in analog low pass filters, though you will likely want to perform additional digital filtering in your control. The control calculations in the prototype worked quite well with a sampling rate of 50 samples per second for each gyro and accelerometer.

Radio/servo input/output pulses

The interface between the GPS-UAV and the servos and the RC receiver is via 5 connection points on the board, labeled EleI, EleO, RudI, RudO, Sell. Signals on these connections are standard RC 5 volt TTL servo pulses. On the prototype board, connections between the board and the radio and servos were accomplished by taking servo extension cables, cutting them in half, and soldering them onto the board. As of this writing it is not clear whether or not the board will be sold with cables already installed. If not, you will need to install your own servo wires into these ports. If the board comes with cables installed, the cables will have “universal” connectors. If the connectors fit your radio, fine. If not, unsolder the cables and install your own.

The recommended method is to use servo extensions for the type of radio and servos that you are using. Cut them in half, and solder them onto the board. Make them long enough to connect between the board and your servos and your receiver. EleI, RudI, and Sell are for connections between the RC receiver and the GPS-UAV, and should have connectors that will plug into your radio. EleO and RudO are for connections between the servos and the GPS-UAV, and should have connectors that will mate with the servo connectors.

Each cable has three conductors, ground, positive, and signal. On a Futaba cable, ground is black, positive is red, and signal is white. The cable wires should be soldered to the board in that order, ground nearest to the edge of the board, positive next, and finally signal. The board ties the positives from all 5 cables together, and ties the grounds from all cables together.

Signal pin connections to the CPU are as follows:

RudI	Pin 26: RB5
RudO	Pin 13: RC2
EleI	Pin 25: RB4
EleO	Pin 12: RC1
Sell	Pin 21: Int0 and Pin 22: Int1

RudI and EleI are rudder and elevator inputs intended to be used as interrupt-on-change signals on RB5 and RB4. This provides a convenient way to measure pulse width.

Sell is a “selection” input used to provide commands to the GPS-UAV via the RC radio. In the prototypes, the selection input came from the throttle channel, and was used to adjust the amount of augmentation in the augmented mode, as well as to select the various controller modes. Of course, you are free to use it any way you would like, but you will likely want to use it for something.

Note that Sell is connected to two CPU pins, both Int0 and Int1. There are two reasons for this. There are only 4 interrupt-on-change pins on the 18F2520, and 2 of them are multiplexed with the PGD and PGC pins, which are used to program and debug the firmware. Also, it was felt useful to provide a high-priority interrupt for the Sell signal, since on the prototype Sell served a critical function. Because the IntX pins are edge triggered, a pair of them is needed to provide an interrupt-on-change functionality.

It is recommended that you enable the built-in pull up resistors that the 18F2520 provides on the B port, because separate resistors are not provided on the board.

There is no direct physical connection between RudI and RudO, and EleI and EleO, on the board. Manual control of the servos from the radio must be done as a “pass-through” in software, but this can be done simply and reliably with a high priority interrupt that will continue to work even if the rest of your firmware fails. The reason the decision was made to

use firmware rather than hardware to provide manual control was to simplify the hardware circuitry and to avoid unnecessary weight.

The GPS-UAV can control two servos. If you are using additional channels, you can still use them in conjunction with the GPS-UAV by making a direct connection between the radio and the extra servos. This is also a useful debugging technique. For example, while you are debugging one axis of the control you can provide manual control for the other axis with a direct connection between radio and servo.

Signals are standard servo pulses, with a pulse width of between 1 millisecond and 2 milliseconds, and approximately 20 milliseconds between pulses. Between pulses, the signal level is a TTL low, and the pulse is a TTL high. Of course, you can measure the characteristics of your own RC radio using the GPS-UAV.

Be careful in designing the duty cycle of pulses for controlling the servos. The approximately 20 millisecond delay between pulses is deliberate, and is a result of the way that the original designers of RC radios used time multiplexing to control several servos over a single radio channel. As a result, there is a “pulse stretcher” within the servos. If you eliminate the delay, or significantly reduce it, the pulse stretchers in the servos will misoperate as well as create a short on the power supply. Therefore, you cannot use the PWM controllers in the 18F2520 in a simple-minded fashion to create a continuous stream of closely spaced pulses. Rather, you will need to fill in a sequence of “null” pulses between the ones that you actually want.

Be careful in your firmware with gains and/or filtering to make sure that there is not excessive servo “chatter”, which will result in excessive servo currents and heating.

ICSP header

The GPS-UAV includes an ICSP header which is used for programming and debugging. If you use the MPLAB compatible ICD2 from Spark Fun, simply connect the cable that comes with it directly between the ICD2 and the GPS-UAV. If you use something else, you may need to use an adapter. Spark Fun sells an adapter between ICSP and RJ11.

It is not recommended that you try to power the GPS-UAV from the ICD2.

Pin assignments on the ICSP header are as follows. Pin 1 is closest to the ICSP label by the connector:

1	VPP
2	VCC
3	GND
4	PGD
5	PGC
6	N/C

LEDS

The GPS-UAV has three surface mounted LEDs, one yellow LED connected to the GPS, and two connected to the CPU, a green one (Stat1) and a red one (Stat2). The yellow LED on the GPS will flash whenever the GPS is producing valid locations. The green and the red LEDs are useful for debugging. However, it is recommended that when you are ready for flight, you should mount the board inside your aircraft, in which case you will not be able to see the LEDs. Once you have your firmware debugged, you can signal status by deflecting the

control surfaces. For example, while the board is zeroing the gyros and waiting for the GPS, it could “waggle” the rudder, and then stop when the process is complete.

Stat1 (green)	RA4
Stat2 (red)	RC0

Switches

Two switches connected between pull up resistors and ground are provided for setting options, debugging, or operation, or whatever you like. They are labeled SR1 and SR2 with the idea that in the final version of your software, you might use them as servo reversing switches. They were not used during the development of the first prototype. Note that two switches are not enough to describe all combinations of board orientations and servo directions. If you decide to use the switches for servo reversing, it is suggested that you select a single preferred board orientation, or account for board orientation in your firmware.

SR1	RC5
SR2	RC4

Power, size, and weight

There are two on-board voltage regulators, one to provide 5 volts to the CPU, gyros, accelerometers, and LEDs, the other to provide 3.3 volts to the GPS. The board provides signal voltage level shifting between the CPU and the GPS.

Only the power for the CPU and GPS flows through the voltage regulators. The power for the radio and the servos flows directly to them without going through the regulators. Total current load on a 4.8 volt battery is about 130 milliamps when the servos are steady.

There is a diode, D1, in series with J1, which you may optionally bypass by soldering jumper PJ1. When the board is operating from a 4.8 volt NiCad, the 5 volt regulator will be dropped out. Bypassing the diode will let you run a little longer. Also, once you are done using a separate power supply for debugging and are ready to use batteries only, you can eliminate the dropout voltage of the 5 volt regulator by bypassing it by soldering jumper PJ2.

Do not apply more than 5.5 volts to the board if you bypass the voltage regulator with PJ2.

The positive voltage of the servo connection, pin 2, of all 5 of the servo connectors are all tied together and to the external power connection, J1. Power for the CPU and GPS then passes through the diode D1 and power switch S1. Under most situations, you will probably want to just leave the power switch S1 in the on position.

As of this writing, it was not known whether or not the board will be populated with J1. If not, you may want to select a connector to mate with a convenient power supply.

There are two recommended ways of powering the board, either by providing approximately 6 volts DC through header J1 or by backfeeding through the servos from the radio receiver that is being powered by a 4.8 volt NiCad. The backfeeding method works because power and ground is connected on all servo connections. Power is provided to the radio directly by the battery, and appears on all servo connectors on the radio. Power flows from the radio to the board on the connections between the radio and the board, where it becomes available for the board and the servos connected to the board. Any servos connected directly to the radio receive power directly from the radio.

Use one method or the other for providing power, but not both at the same time.

It is recommended that you use 6 volts external power on J1 only when you are debugging without the radio or servos connected. When you have the radio and servos connected to the board, you should supply power with the battery that is recommended for your radio, such as a 4.8 volt NiCad. This is the method that was used to develop the first prototype.

Using external power is useful in the early stages of development, when you may be spending long periods of time with power applied to the board, particularly when you are developing an interface to the GPS. For external power without radio or servos connected, supply approximately 6 to 9 volts DC to J1. Another option for debugging without radio or servos is to simply plug a NiCad directly into EleI, RudI, or Sell, without any power supply connected to J1.

When you are ready to test with servos and a radio interface, connect the radio to EleI, RudI, and Sell. Connect the servos to EleO and RudO. Connect a 4.8 volt NiCad to the radio. The board will tie power for the radio and the servos together and tap off the power that it needs.

The board has not been tested with Lithium Polymer batteries, so you are on your own if you want to go with LiPo, but here are some considerations: The use of a single cell LiPo will limit the useful range of the gyros because of the internal 2.5 volt reference. The use of a two cell LiPo will apply approximately 7.4 volts to the radio and servos, which exceeds the rating of many commonly used products, and will cause a control voltage level shift between the radio, servos, and CPU I/O pins, which will be running at 5 volts. If you want to use LiPo, you would probably do well to use 2 cells and a battery eliminator circuit to drop down to 5 volts.

Board size is 1 and 5/8 inches by 5 inches, and will fit quite nicely inside many sailplane models, including the one in which the prototype was developed, a "Gentle Lady".

The total weight of the board and antenna is just less than 2 ounces. If weight is an issue, you may want to consider using a light weight battery and servos.

Recommended accessories

In addition to the GPS-UAV, there are several recommended accessory items that you can buy from Spark Fun:

- Antenna GPS Embedded SMA. This item is highly recommended as the antenna you will use in your sailplane. You might want to buy 2 of them, in case you break the wire off one of them from frequent handling.
- MPLAB Compatible ICD2 with USB and RS232. This item is highly recommended. You will need something to program the CPU, and this is a good choice. It is the programming/debugging interface used to develop the prototype. An assembler/linker/debugger software tool is available for free from Microchip that will be sufficient for firmware development. The prototype was developed using this combination, without benefit of a C compiler. If you prefer, there are commercially available C compilers that are compatible with the ICD2, with price tags of several hundred dollars.
- Coin Cell Battery – 12 mm. This is an entirely optional item, but it is rather inexpensive, lightweight, and will save a great deal of time at the flying field by enabling a warm start for the GPS.
- ET301 SiRFIII Evaluation Kit. This is an entirely optional item. You can get by without it, but having it will save a great deal of development time by allowing you to play around with the NMEA interface through your computer. The kit includes a battery, power supply, and magnetic mount antenna.

- Antenna GPS 3V Magnetic Mount SMA. If you do not buy the GPS evaluation kit, this item is highly recommended, it will greatly speed debugging. One way that it was used during the development of the prototype was to attach it to the roof of a car, and then testing the firmware in motion by driving the car around.