

GPS Vehicle Tracker

This article describes a simple way to track the location of a vehicle and identify whether it has an authorized driver. This has many uses such as finding stolen vehicles, creating automatic vehicle logs, and managing a fleet of trucks and buses.

Coordinates, in degrees latitude and longitude, are transferred via a standard mobile telephone "text" message, which can then be used to determine the exact location of the vehicle.

The code for this project is available for download:

- [EPD-v2-01-3E.BSP](#): Main program for the Basic Stamp.
- [1WIRE-ID.BSP](#): A program to 'read' the serial number of the DS1990A iButton.

Ingredients

The project uses the following building blocks:

1. **GPS receiver** (Garmin GPS 35). This provides information about the location of the vehicle.
2. **GSM module** (Siemens TC35). The GSM module is used to send SMS (text messages) to a pre-programmed mobile number.
3. **iButton** (DS1990A). The iButton identifies authorized drivers.
4. **Parallax Basic Stamp** (BS2P) in conjunction with a BASIC STAMP "Board of Education" development board. (www.parallaxinc.com)
5. **Software:**
 - EPD-v2-01-3E.BSP: Main program for the Basic Stamp.
 - 1WIRE-ID.BSP: A program to 'read' the serial number of the DS1990A iButton.

Block Diagram

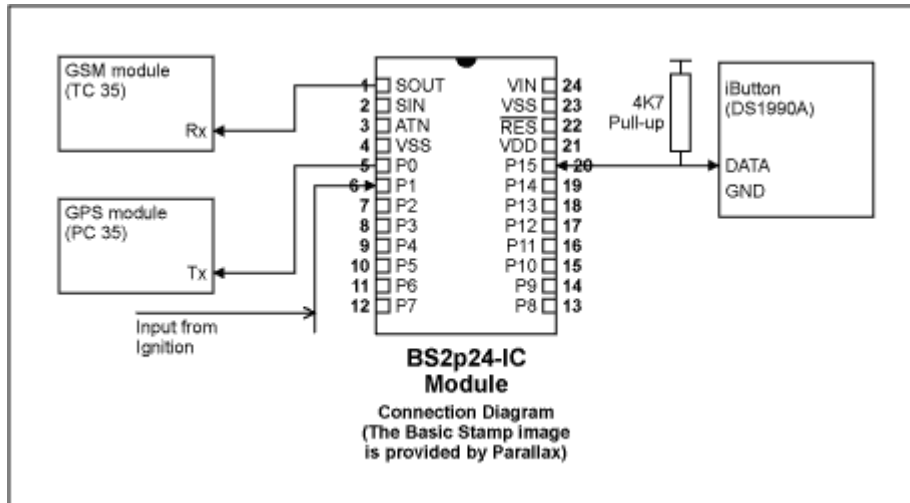


Figure 1. Block diagram

The GPS receiver provides information in 'NMEA' format. The latitude and longitude can be extracted from the '\$GPRMC' sentence (in form of "ABC degrees North", "DEF degrees West").

The GSM module is used to send SMS messages (text messages) to a pre-programmed mobile phone.

The DS1990A iButton (also known as "serial number iButton") is used as a means of identification or "Key." It contains a unique factory programmed 6-byte ID with 2 bytes of CRC (error correction). The key codes are read-only. The system will accept two iButton keys, allowing a spare key, or second driver. More key codes can be easily added.

Operation

On start-up, the BASIC STAMP continuously scans for two things:

- an iButton to be momentarily connected to the input
- the ignition being active

Normal operation would be for an 'authorized' iButton to be connected to the port and then the car is started within 25 seconds.

Intrusion is detected by the following conditions:

- If the ignition is detected to be active without a valid iButton within 25 seconds of starting the car.
- If no iButton is connected within 25 seconds of starting the car, this will also be an intrusion state.

Upon establishing that an intrusion has occurred, the BASIC STAMP looks for the information from the GPS receiver. It waits for the string '\$GPRMC' to appear on the serial output of the GPS receiver.

- It then sends an SMS message to a pre-programmed mobile number with the coordinates extracted from the 'GPRMC' sentence received from the GPS.
- This process is repeated every two minutes to keep the owner informed of the car's location.

Putting It All Together

The core of this project is the BASIC STAMP. It interfaces to the GPS receiver, the GSM modem, the iButton and the ignition connection from the vehicle.

In this project the Ignition line is used to detect if the car has been started. Other indicators can be used (e.g., ultrasonic transducer or other forms of intrusion detectors).

- The GPS receiver's output is connected to P0 of the basic stamp.
- The GSM module's receiver input is connected to 'Sout' or Pin 1 of the Basic Stamp.
- P1 of the basic stamp is connected to the ignition connection. A high state indicates that the ignition is on and vice versa.
- P15 is used for the interface to the iButton.

Note that the BS2P in the BASIC STAMP range will support the 1-wire interface for the iButtons, previous variants will not support the interface.

Entering Your Mobile Number

To program a mobile phone number in the code, enter it in the line indicated in the program as shown below:

```
=====
DEBUG "+44*****" ' Enter the destination mobile phone number here
=====
```

It must be in the format:

+ Countrycode (using UK's code in the example above) followed by the mobile's number omitting the leading zero from the number.

This will be the number that the system will use to send SMS messages with the location of the car.

Entering the iButton Serial Numbers

Enter the iButton serial numbers of the iButtons that are to be used. If only a single iButton is used, then duplicate the serial number in the slots for the 2nd iButton.

The iButton key codes are entered as illustrated below. In this case, example iButton numbers have been entered. Replace the numbers following the '\$' symbols. The serial numbers are represented in hexadecimal.

A separate program can be supplied that will display the serial number of specific iButtons. The serial numbers can then be entered into the final code that will be downloaded to the BASIC STAMP.

'ENTER THE FIRST iBUTTON Serial Number is the lines below'

```
IF romData(0) <> $03 THEN CheckSecond 'First Byte
IF romData(1) <> $F0 THEN CheckSecond
IF romData(2) <> $BC THEN CheckSecond
IF romData(3) <> $08 THEN CheckSecond
IF romData(4) <> $10 THEN CheckSecond
IF romData(5) <> $00 THEN CheckSecond
IF romData(6) <> $00 THEN CheckSecond
IF romData(7) <> $30 THEN CheckSecond 'Eighth Byte
```

GOTO ButtonOK

CheckSecond:

'ENTER THE SECOND/Spare iBUTTON Serial Number is the lines below'

```
IF romData(0) <> $01 THEN BadButton 'First Byte
IF romData(1) <> $68 THEN BadButton
IF romData(2) <> $21 THEN BadButton
IF romData(3) <> $24 THEN BadButton
IF romData(4) <> $08 THEN BadButton
IF romData(5) <> $00 THEN BadButton
IF romData(6) <> $00 THEN BadButton
IF romData(7) <> $31 THEN BadButton 'Eighth Byte
```

After making these modifications, the system should be able to track any vehicle.

The code for this project is available for download:

[EPD-v2-01-3E.BSP](#): Main program for the Basic Stamp.

[1WIRE-ID.BSP](#): A program to 'read' the serial number of the DS1990A iButton.

A similar version of this article appeared in the November 2002 issue of *EPD* magazine.

A354, December 2002