Adafruit Mini GPS PA1010D UART Guide

CMPT 433: Embedded Systems

Team BeagleDashCam
Elaine Abraham
Diego Buencamino
Kai Feng (Kevin) Chen
Ngan Ngoc (Lynn) Nguyen

Spring 2024

This guide has been tested on:

- BeagleBone (the **target**): Debian 11.8
- Host Laptop (the **host**): Debian 11.9

Tested on Linux Kernel 5.10

This document guides the user through:

1. Introducing the UART communication protocol
2. Wiring up the GPS module
3. Initializing the UART interface on the target
4. Reading the GPS module's data using the command line
5. Using a C program to access the GPS module's data

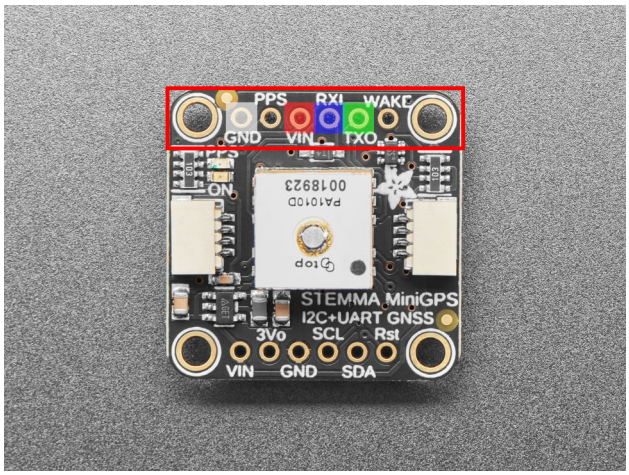
1 The UART Protocol

UART, or *universal asynchronous receiver-transmitter*, is a communication protocol used for sending or receiving serial data between two electronic devices. UART communication consists of a *transmitter*, which serializes parallel data into a single byte stream, and a *receiver*, which then reassembles the incoming bits into parallel data. Devices transmitting through UART agree on a specific rate for data transmission (also called a *baud* rate).

The BeagleBone Green has dedicated pin-outs for up to five serial connections, which are all accessible within Linux. Note that the GPS module used in this guide can be wired up to only four of these channels (one of the five can only transmit, not receive).

2 Wiring the Module Up

You will need 4 jumper cables for this part of the setup. As mentioned in the previous section, the module can be wired up to any of the four UART channels present on the board. **The pin header strips (not shown in the image) should already be soldered in.** Otherwise, consult with the course instructor (or solder it yourself).



1. Attach the module to the breadboard, making sure the pins boxed in red are the ones being inserted.
2. Connect the **GND** pin (highlighted in white) to any ground pin (`P9_1` for example).
3. Connect the **VIN** pin (highlighted in red) to `P9_7` or `P9_8` (the `SYS_5V` pins).
4. Connect the **RXI** pin (highlighted in blue) to a **TXD** pin corresponding to one of the UART channels on the table below.
5. Connect the **TXO** pin (highlighted in green) to the **RXD** pin on the same UART channel as your TXD pin.

UART Channel	**TXD** Pin	**RXD** Pin
`UART1`	`P9_24`*	`P9_26`*
`UART2`	`P9_21`	`P9_22`*
`UART4`	`P9_13`*	`P9_11`*
`UART5`	`P8_37`	`P8_38`

*These pins conflict with some of the modules on the Zen Cape Red.

`P9_22` is used for the Zen Cape Red's buzzer module.

`P9_24` and `P9_26` are used by some part of the accelerometer.

`P9_13` and `P9_11` are used by the tri-colour LEDs.

3 Initializing the UART Interface

The TXD and RXD pins you used in the previous section will determine which pins will be configured at this step. For the rest of this guide, `P9_13` and `P9_11` (the pins for `UART4`) will be used. Make sure to configure the pins that your module is plugged into.

6. On the target, configure the pins to UART:

```
(target)$ sudo config-pin p9.13 uart
(target)$ sudo config-pin p9.11 uart
```

You should now be able to access the GPS module through Linux.

4 Reading the GPS Module's Data

A previous iteration of this GPS guide outlined steps to load an additional overlay to get UART to work on the BeagleBone. The version of software used on the BeagleBone for Spring 2024 did not require us to load this overlay.

If the module is wired correctly, a solid green light will appear on the face of the module. A blinking red light alongside this indicates that your module has *hooked* onto satellites and is currently receiving location data.

7. See all the serial channels present on the BeagleBone:

```
(target)$ ls /dev/ttyS*
```

8a. Access the GPS module's data:

```
(target)$ cat /dev/ttyS4
```

The GPS module outputs data in the form of NMEA sentences:

```
$GNGGA,002153.000,3342.6618,N,11751.3858,W,1,10,1.2,27.0,M,-34.2,M,,0000*5E
$GPGSA,A,3,07,02,26,27,09,04,15, , , , , ,1.8,1.0,1.5*33
$GPGSV,2,1,07,07,79,048,42,02,51,062,43,26,36,256,42,27,27,138,42*71
$GPGSV,2,2,07,09,23,313,42,04,19,159,41,15,12,041,42*41
$GNRMC,161229.487,A,3723.2475,N,12158.3416,W,0.13,309.62,120598, ,*10
$GPVTG,309.62,T, ,M,0.13,N,0.2,K,A*23
...
```

Sentences of interest are the ones that start with `$GNGGA`, as these are the ones that contain coordinate data (including latitude, longitude, and direction). **Parsing the coordinate data is required, but will not be covered in this guide. NMEA sentences do not use the same coordinate formatting that is normally used in software such as Google Maps.**

8b. Access the GPS module's data, but only ones that begin with `$GNGGA`:

```
(target)$ cat /dev/ttyS4 | grep \$GNGGA
```

5 Reading the GPS Module's Data using C

Interfacing with the BeagleBone's UART interface can be done using the `termios.h` library, as outlined in this article by Geoffrey Hunter, and referenced in a previous guide. The article provides a detailed explanation for these flags and how they affect communication through your chosen UART channel.

```
#include <termios.h>

int serial_port;

void GPS_init() {
  runCommand("config-pin p9.11 uart");

  serial_port = open("/dev/ttyS4", O_RDWR);
  if (serial_port < 0) {
    printf("Error %i from open: %s\n", errno, strerror(errno));
  }

  struct termios tty;
  if(tcgetattr(serial_port, &tty) != 0) {
    printf("Error %i from tcgetattr: %s\n", errno, strerror(errno));
  }

  tty.c_cflag &= ~PARENB;
  tty.c_cflag &= ~CSTOPB;
  tty.c_cflag &= ~CSIZE;
  tty.c_cflag |= CS8;
  tty.c_cflag &= ~CRTSCTS;
  tty.c_cflag |= CREAD | CLOCAL;
  tty.c_lflag |= ICANON;
```

```
    tty.c_lflag &= ~ECHO;
    tty.c_lflag &= ~ECHOE;
    tty.c_lflag &= ~ECHONL;
    tty.c_lflag &= ~ISIG;
    tty.c_iflag &= ~(IXON | IXOFF | IXANY);
    tty.c_iflag &= ~(IGNBRK|BRKINT|PARMRK|ISTRIP|INLCR|IGNCR|ICRNL);
    tty.c_oflag &= ~OPOST;
    tty.c_oflag &= ~ONLCR;
    tty.c_cc[VTIME] = 10; // deciseconds
    tty.c_cc[VMIN] = 0;   // deciseconds

    // Baud Rate
    cfsetspeed(&tty, B9600);

    if (tcsetattr(serial_port, TCSANOW, &tty) != 0) {
      printf("Error %i from tcsetattr: %s\n", errno, strerror(errno));
    }
  }

```

After your settings have been configured, you can read data from the UART channel using the file descriptor you have configured.

```
  char* GPS_read() {
    static char read_buf[255];

    int n = read(serial_port, &read_buf, sizeof(read_buf));

    if (n > 0) {
      read_buf[n] = '\0';
      return read_buf;
    }
    else {
      static char empty_string[] = "";
      return empty_string;
    }
  }

```

6 Troubleshooting

There's no green light on the GPS module!

- Use the *Wiring the Module Up* section as a reference to double check that your wiring is correct.
- The GPS module will sometimes exhibit weird behaviour when using a 3.3V pin, so using a 5V pin is always recommended.

Nothing shows up when I run `cat /dev/ttyS4`!

- Make sure that you are accessing the correct serial port.
- Make sure that the pins that you are using for the GPS module are configured for UART.

I can't seem to get a *hook* on the GPS!

- The antenna on the GPS module is very sensitive to obstacles. If you have to work indoors, make sure that the module has a clear view of the sky.
- Go outside. The GPS is more reliable outdoors.

The buzzer makes some weird sounds! The tri-colour LEDs are flashing like crazy!

- As mentioned in *Wiring the Module Up*, some of the pins used for UART conflict with some of the Zen Cape Red's modules. This does not affect the functionality of the GPS module, but will affect the functionality of the modules that conflict with it.
- If you need one (or both) of these modules, try using a different UART channel that does not conflict with the module you are trying to use.
- If you are using the UART channel that conflicts with the buzzer, you can pull the jumper adjacent to the buzzer module to silence it.

Acknowledgements

As of Spring 2024, there have been two other guides that walk you through connecting and using a GPS module with your BeagleBone Green:

- GPS-AdaFruit – PA1616S Via UART
- Grove GPS – SIM28 – UART Guide For BBG

We have found that these guides:

- ... were developed for different modules,
- ... used the same pins to connect their respective GPS modules, without acknolwedging the other pinouts available on the BeagleBone for UART communication, or
- ... had unclear wiring instructions.

We are hoping to use this guide to iterate on the two other GPS guides, and make it easier for you to wire up and start using your GPS module.