

Maduino GPRS GPS A7 v1.5 user manual

Maduino GPRS GPS A7 is a main board based on the ATmega328 and GPRS/GSM GPS module A7. Which onboard with **Lipo battery charge circuit** and a **micro SD card holder**. We use the **Arduino pro mini 3.3V 8MHz bootloader** in this board and use the **CP2104** as USB to serial to upload the code by Arduino IDE.

A7 supports quad-band 850/900/1800/1900MHz that covers any GSM network in the world. Simply insert a 2G Micro SIM card, then you are able to enchant things with cellular connectivity.

With this board, you will easy to add text, SMS and data to your project. It is good for your smart home project or GPS tracker and so on.

Features:

BAT Input Voltage: 3.4-4.2V

ATmega328: 8MHz, 32KB flash, 2KB SRAM

Micro SIM connector

Integrated Power Control System

Support AT Command

Quad-band: 850/900/1800/1900Mz

Support GPS

Support GPRS data traffic, the maximum data rate, download 85.6Kbps, upload 42.8Kbps

Support SMS text messaging

Interface: I2C/SPI/UART/18*GPIO

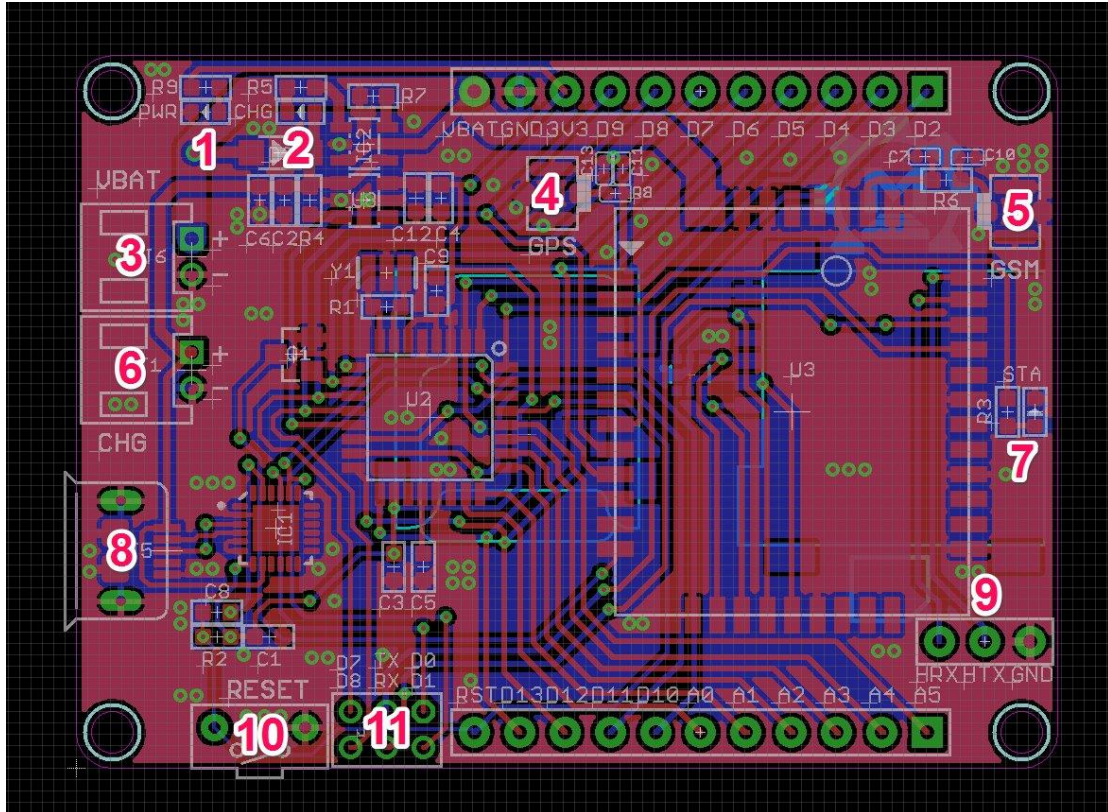
Arduino compatible

Working Temperature: -40 – 85°C

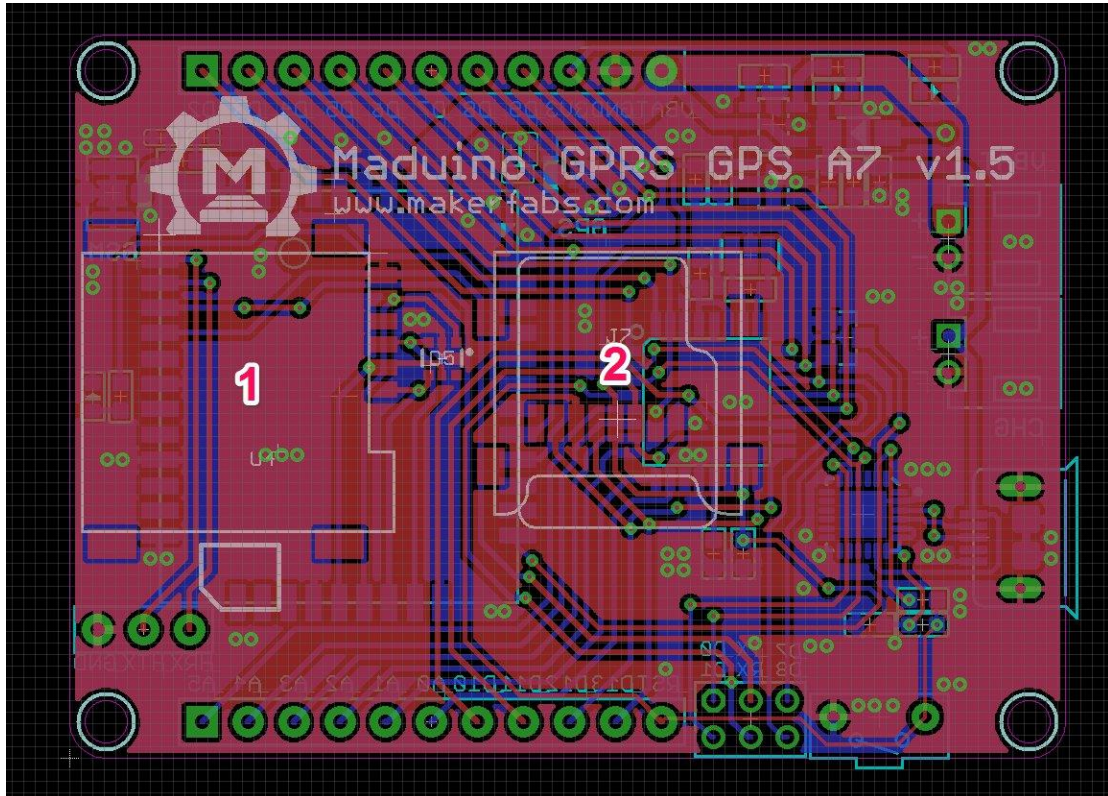
Default baud rate: 115200

Size: 40*55mm

Interface:



1. PWR: Power indicate
2. CHG: Charge indicate
3. VBAT: 3.7V Lipo battery connector
4. GPS: GPS Antenna IPX Interface
5. GSM: GPRS/GSM Antenna IPX Interface
6. CHG: 5V power input, can connect the solar panel to charge the lipo battery.
7. STA: A7 status indicate
8. Micro USB: 5V power input, USB to serial communication
9. A7 serial port for firmware update
10. RESET: Reset button for ATmega328
11. Serial port select (D0, D1 hardware serial port D7,D8 Software serial port)



1. Micro SIM Card holder
2. Micro SD Card holder

Pins usage on Arduino

- D0** - Unused if you select hardware serial port to communicate with A7
- D1** - Unused if you select hardware serial port to communicate with A7
- D2** - Unused
- D3** - Unused
- D4** – SD Card CS PIN
- D5** – Low power control of A7 Low level active
- D6** – Power OFF of A7 High lever active
- D7** - Used if you select software serial port to communicate with A7
- D8** - Used if you select software serial port to communicate with A7
- D9** – Power ON of A7 High lever active (High level more than 3S then Set LOW level)
- D10** - Unused
- D11** - Unused
- D12** - Unused
- D13** - Unused
- D14(A0)** - Unused
- D15(A1)** - Unused

D16(A2) - Unused

D17(A3) - Unused

D18(A4) - Unused

D19(A5) - Unused

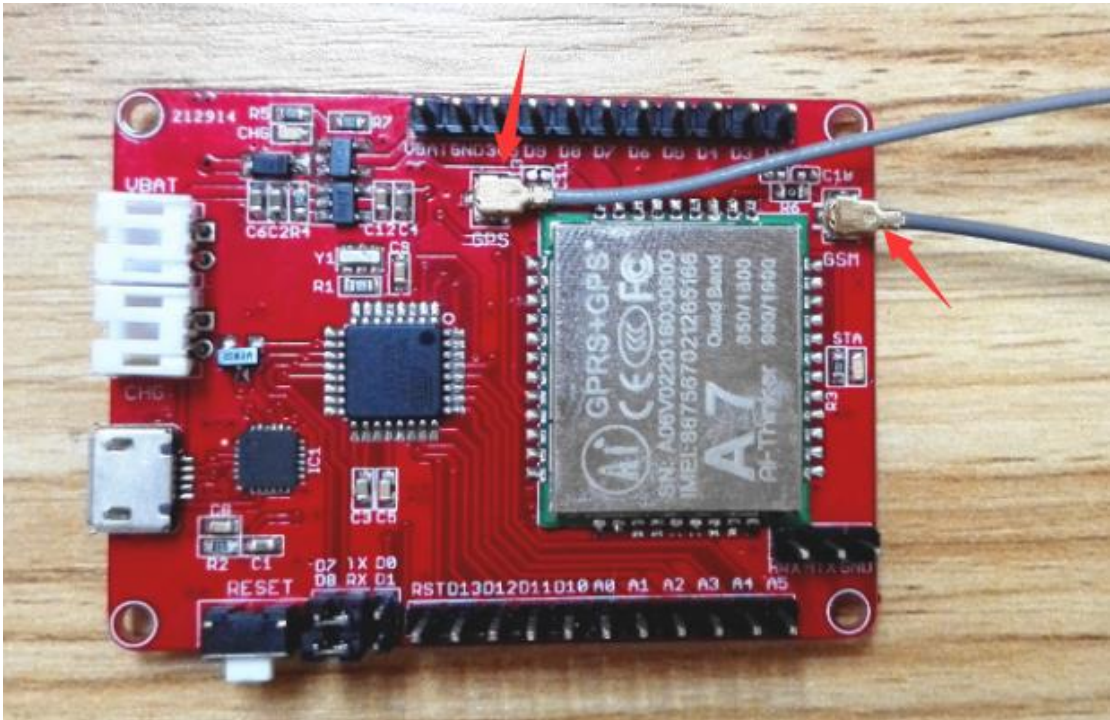
Usage:

Hardware installation

1. Insert an Micro SIM card and Micro SD card



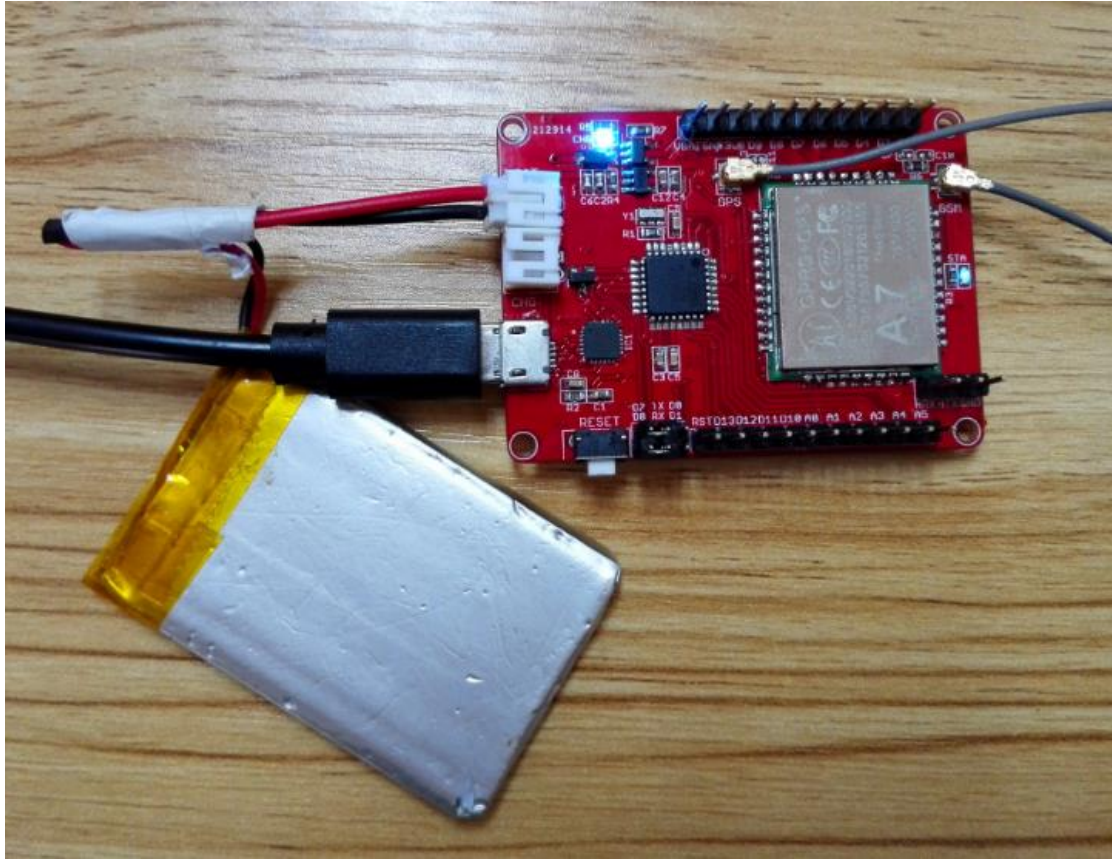
2. Connect the Antenna



3. Connect the 3.7V Lipo battery



4. Connect the USB Cable



5. A7 GPRS HTTP Test

Please select the board: Arduino pro mini 3.3V 8MHz

Upload the code AT GPRS HTTP Test.ino

```
Arduino 1.6.4
File Edit Sketch Tools Help
A7_GPRS_HTTP_Test
#include <SoftwareSerial.h>

SoftwareSerial mySerial(7, 8);
#include<stdio.h>
#include<string.h>
#define DEBUG true

void setup()
{
  Serial.begin(9600);
  mySerial.begin(9600); //Default serial port setting for the GPRS modem is 9600 8-N-1
  delay(1000);
  int pon=9;
  int poff=6;
  int lowp=-5;
  // initialize digital pin 13 as an output.
  pinMode(pon, OUTPUT);
  pinMode(poff, OUTPUT);
  pinMode(lowp, OUTPUT);
  digitalWrite(poff, LOW);
  digitalWrite(lowp, HIGH);
  digitalWrite(pon, HIGH);
  delay(3000); // wait for a second
  digitalWrite(pon, LOW); // turn the LED off by making the voltage LOW
  delay(1000);
  baudrate();
  Serial.println("After 3s, test begin!");
}

Done uploading

Global variables use 527 bytes (25%) of dynamic memory, leaving 1,521 bytes for local variables. Maximum is 2,048 bytes.

48 Arduino Pro or Pro Mini, ATmega328 (3.3V, 8 MHz) on COM3
```

Open the serial monitor and set the baud rate as 9600, see the print information.

```
COM3
Send
AT+CIPSTART="TCP","gtech.co",80
CONNECT OK
OK
AT+CIPSEND=80
> GET http://gtech.co/gtech.php?hello HTTP/1.0
HOST:gtech.co
AT+CIPCLOSE
OK
+CIPRCV:186,HTTP/1.1 200 OK
Date: Fri, 07 Apr 2017 06:AT+CCID
+CCID:89860063191507135572
OK
AT+CREG?
+CREG: 1,1
OK
AT+CGAII=1
OK
AT+CGACT=1,1
OK
```

Autoscroll No line ending 9600 baud

6. Get GPS information

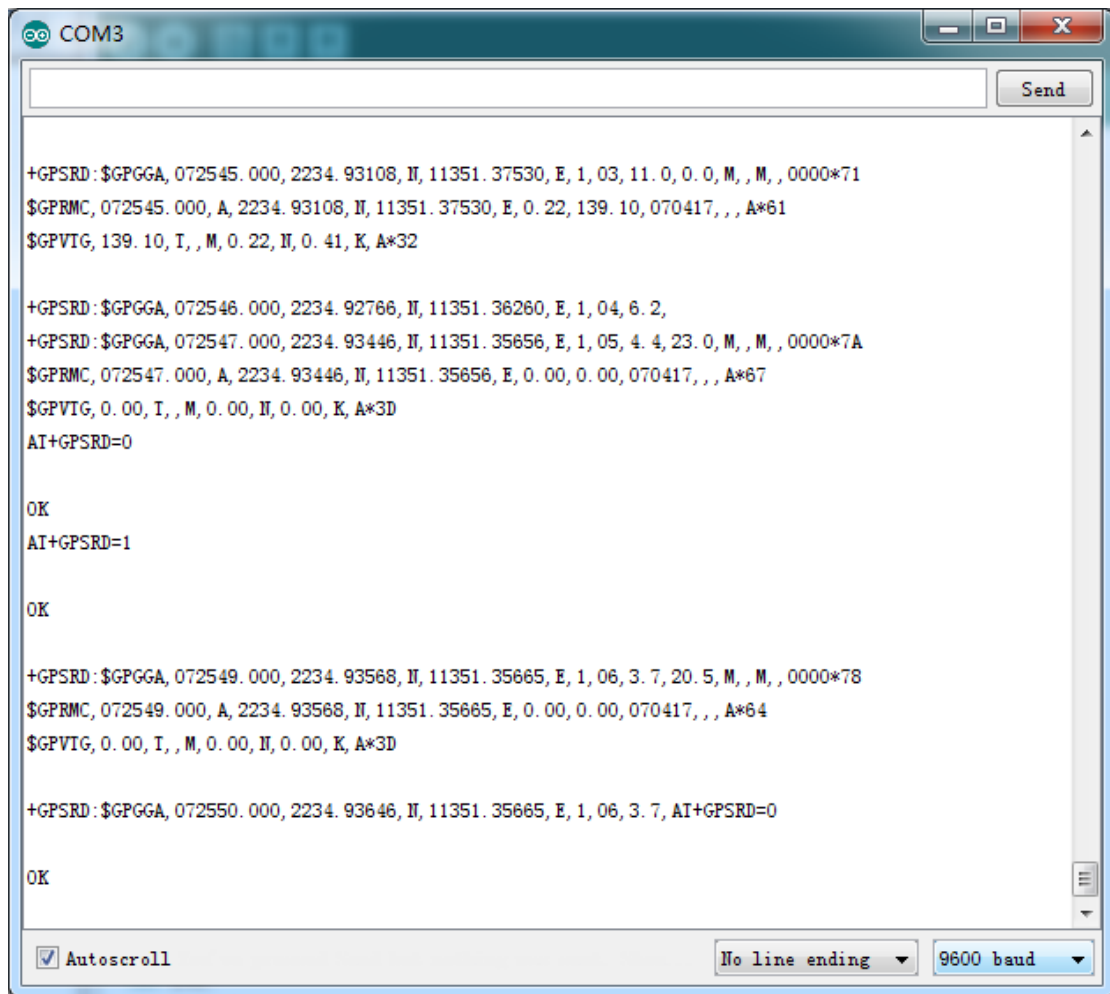
AT Command:

AT+GPS=1 //open GPS (NEMA information output from GPS_TXD PIN, but in this board not breakout this pin, please set GPS info output from AT Command serial port)

AT+GPS=0 //close GPS

AT+GPSRD=N //NEMA information output N second form serial port

AT+GPSRD=0 // Close the GPS information output form serial port



The screenshot shows a serial terminal window with the title 'COM3'. The window contains several lines of text, which are GPS NMEA sentences and control commands. The text is as follows:

```
+GPSRD:$GPGGA,072545.000,2234.93108,N,11351.37530,E,1,03,11.0,0.0,M,,M,,0000*71
$GPRMC,072545.000,A,2234.93108,N,11351.37530,E,0.22,139.10,070417,,A*61
$GPVTG,139.10,I,,M,0.22,N,0.41,K,A*32

+GPSRD:$GPGGA,072546.000,2234.92766,N,11351.36260,E,1,04,6.2,
+GPSRD:$GPGGA,072547.000,2234.93446,N,11351.35656,E,1,05,4.4,23.0,M,,M,,0000*7A
$GPRMC,072547.000,A,2234.93446,N,11351.35656,E,0.00,0.00,070417,,A*67
$GPVTG,0.00,I,,M,0.00,N,0.00,K,A*3D
AI+GPSRD=0

OK
AI+GPSRD=1

OK

+GPSRD:$GPGGA,072549.000,2234.93568,N,11351.35665,E,1,06,3.7,20.5,M,,M,,0000*78
$GPRMC,072549.000,A,2234.93568,N,11351.35665,E,0.00,0.00,070417,,A*64
$GPVTG,0.00,I,,M,0.00,N,0.00,K,A*3D

+GPSRD:$GPGGA,072550.000,2234.93646,N,11351.35665,E,1,06,3.7,AI+GPSRD=0

OK
```

At the bottom of the window, there are three controls: a checked checkbox labeled 'Autoscroll', a dropdown menu set to 'No line ending', and another dropdown menu set to '9600 baud'.

7. SD card test

Open the demo code Files->Examples->SD->ReadWrite

