1502177743
VÝROBNÍ ČÍSLO

GPS modul NEO-7M

1. POPIS

Tento malý a kompaktní modul je schopen určit vaši polohu prostřednictvím satelitní navigace GPS.
Lze ho připojit k PC pomocí TTL převodníku, zabudovaného USB konektoru nebo za pomoci Arduina. Pokud uživatel zvolí připojení pomocí TTL převodníku, má možnost vyzkoušet si aplikaci U-Center, která má grafické uživatelské rozhraní a podporuje velké množství funkcí a přehledně vypisuje důležité informace o poloze.

Základní charakteristika:

- Kompaktní rozměry
- Připojení pomocí TTL, micro USB nebo Arduina
- Určení polohy díky satelitní navigaci
- Lze použít s aplikací U-Center
- Připojení aktivní antény (konektor SMA)
- Pro přesné údaje je doporučeno umístit modul na otevřené prostranství

2. SPECIFIKACE

Hlavní čip	NEO-7M-C	**Impedance TXD/RXD**	510 Ω
Oscilátor	TCXO	**Trackovací citlivost**	-162 dBm
Baud rate	9600 bps	**Komunikační protokol**	NMEA/UBX
Obnovovací frekvence	až 10 HZ	**Pracovní teplota**	-40 až 80 °C
Pracovní napětí	2,7 až 5 VDC	**Rozměry (mm)**	40 x 25 x 13
Proud	35 mA	**Hmotnost**	21 g

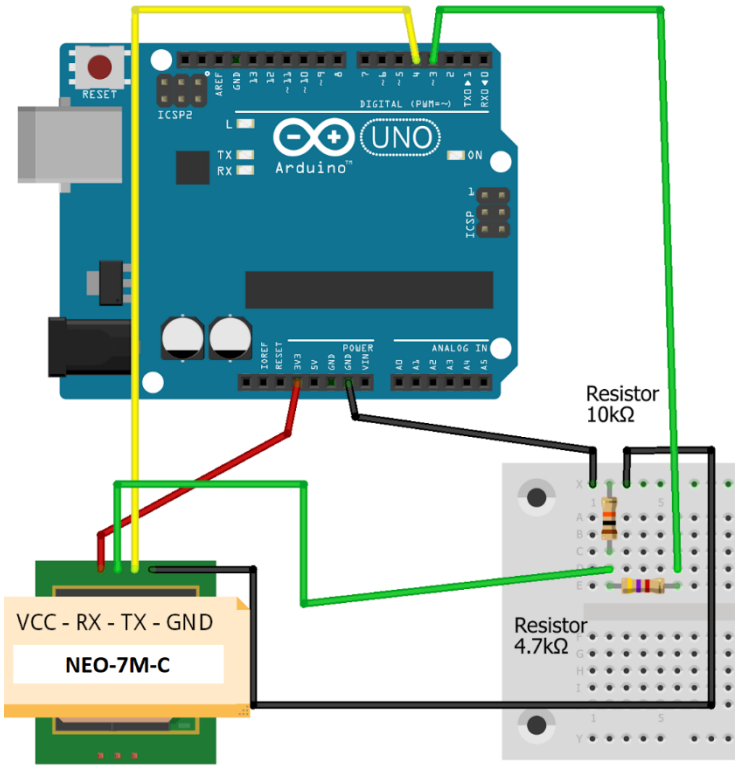
3. ZAPOJENÍ

Připojení s USB TTL převodníkem

TTL převodník najdete na našem e-shopu pod kódem 1421353514. Po připojení k PC doporučujeme nainstalovat aplikaci U-Center, která usnadňuje práci s modulem a poskytuje podstatné informace o poloze.

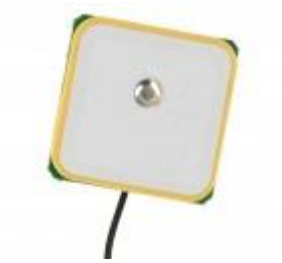
Modul lze také připojit pomocí micro USB. V tomto případě není potřeba TTL převodník.

Připojení k Arduinu



Pasivní anténa

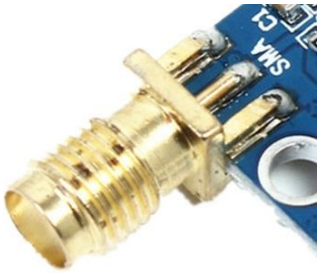
Pasivní anténa je již součástí modulu. Je připojena přímo k plošnému spoji modulu, tudíž není odnímatelná. Pasivní anténa je naprosto dostatečná, pokud je modul umístěn v otevřeném prostranství nebo například na palubní desce auta, kdy modul s anténou míří směrem k přednímu oknu. Pokud byl modul například uvnitř místnosti, data by byla velice nepřesná kvůli špatnému spojení se satelitní navigací. Pasivní anténa je také relativně náchylná na signálové rušení. Tudíž v oblasti s vysokou koncentrací signálů (jiné GPS moduly, WiFi, atd.) mohou být data rovněž zkreslena.



Příklad pasivní antény

Aktivní anténa

Aktivní anténa již není součástí balení, uživatel si ji však může dokoupit. Na modulu je již zabudován konektor pro připojení přídavné antény (konektor SMA). Anténa by měla být anténa opět umístěna v otevřeném prostranství. Pokud je zapojena aktivní anténa, pasivní anténa je nečinná. Externí anténa může být dlouhá cca 3m (může být i delší, ale záleží na úrovni stínění vodiče). Pokud je aktivní anténa v provozu proudový odběr modulu je vyšší až o 20 mA a příkon se může zvýšit až o 60 mW. Kompenzací je ovšem odolnost vůči rušení jinými signály a vnějšími vlivy. Pro přesnější měření údajů o poloze v náročnějších podmínkách je tedy doporučeno použít aktivní anténu.



SMA konektor

4. UKÁZKA PROGRAMU

Pro úspěšné nahrání tohoto examplu do Arduina je nutné stáhnout knihovnu TinyGPS++.h.

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

static const int RXPin = 4, TXPin = 3;
static const uint32_t GPSBaud = 9600;
// The TinyGPS++ object
TinyGPSPlus gps;
// The serial connection to the GPS device
SoftwareSerial ss(RXPin, TXPin);
void setup()
{
  Serial.begin(115200);
  ss.begin(GPSBaud);
  Serial.println(F("FullExample.ino"));
  Serial.println(F("An extensive example of many interesting TinyGPS++ features"));
  Serial.print(F("Testing TinyGPS++ library v. ")); Serial.println(TinyGPSPlus::libraryVersion());
  Serial.println(F("by Mikal Hart"));
  Serial.println();
  Serial.println(F("Sats HDOP Latitude   Longitude   Fix  Date       Time     Date Alt    Course Speed
Card  Distance Course Card  Chars Sentences Checksum"));
  Serial.println(F("          (deg)     (deg)      Age                      Age  (m)    --- from GPS ----  ---- to London  ---
-  RX    RX       Fail"));
  Serial.println(F("----------------------------------------------------------------------------------------------------------------------------------------
--"));
}
void loop()
{
  static const double LONDON_LAT = 51.508131, LONDON_LON = -0.128002;
  printInt(gps.satellites.value(), gps.satellites.isValid(), 5);
  printInt(gps.hdop.value(), gps.hdop.isValid(), 5);
  printFloat(gps.location.lat(), gps.location.isValid(), 11, 6);
  printFloat(gps.location.lng(), gps.location.isValid(), 12, 6);
  printInt(gps.location.age(), gps.location.isValid(), 5);
  printDateTime(gps.date, gps.time);
  printFloat(gps.altitude.meters(), gps.altitude.isValid(), 7, 2);
  printFloat(gps.course.deg(), gps.course.isValid(), 7, 2);
  printFloat(gps.speed.kmph(), gps.speed.isValid(), 6, 2);
  printStr(gps.course.isValid() ? TinyGPSPlus::cardinal(gps.course.value()) : "*** ", 6);
  unsigned long distanceKmToLondon =
    (unsigned long)TinyGPSPlus::distanceBetween(
      gps.location.lat(),
      gps.location.lng(),
      LONDON_LAT,
      LONDON_LON) / 1000;
```

```
  printInt(distanceKmToLondon, gps.location.isValid(), 9);
  double courseToLondon =
    TinyGPSPlus::courseTo(
      gps.location.lat(),
      gps.location.lng(),
      LONDON_LAT,
      LONDON_LON);
  printFloat(courseToLondon, gps.location.isValid(), 7, 2);
  const char *cardinalToLondon = TinyGPSPlus::cardinal(courseToLondon);
  printStr(gps.location.isValid() ? cardinalToLondon : "*** ", 6);
  printInt(gps.charsProcessed(), true, 6);
  printInt(gps.sentencesWithFix(), true, 10);
  printInt(gps.failedChecksum(), true, 9);
  Serial.println();

  smartDelay(1000);
  if (millis() > 5000 && gps.charsProcessed() < 10)
    Serial.println(F("No GPS data received: check wiring"));
}
// This custom version of delay() ensures that the gps object
// is being "fed".
static void smartDelay(unsigned long ms)
{
  unsigned long start = millis();
  do
  {
    while (ss.available())
      gps.encode(ss.read());
  } while (millis() - start < ms);
}
static void printFloat(float val, bool valid, int len, int prec)
{
  if (!valid)
  {
    while (len-- > 1)
      Serial.print('*');
    Serial.print(' ');
  }
  else
  {
    Serial.print(val, prec);
    int vi = abs((int)val);
    int flen = prec + (val < 0.0 ? 2 : 1); // . and -
    flen += vi >= 1000 ? 4 : vi >= 100 ? 3 : vi >= 10 ? 2 : 1;
    for (int i=flen; i<len; ++i)
      Serial.print(' ');
  }
  smartDelay(0);
```

```
}
static void printInt(unsigned long val, bool valid, int len)
{
  char sz[32] = "******************";
  if (valid)
    sprintf(sz, "%ld", val);
  sz[len] = 0;
  for (int i=strlen(sz); i<len; ++i)
    sz[i] = ' ';
  if (len > 0)
    sz[len-1] = ' ';
  Serial.print(sz);
  smartDelay(0);
}
static void printDateTime(TinyGPSDate &d, TinyGPSTime &t)
{
  if (!d.isValid())
  {
    Serial.print(F("********** "));
  }
  else
  {
    char sz[32];
    sprintf(sz, "%02d/%02d/%02d ", d.month(), d.day(), d.year());
    Serial.print(sz);
  }

  if (!t.isValid())
  {
    Serial.print(F("******** "));
  }
  else
  {
    char sz[32];
    sprintf(sz, "%02d:%02d:%02d ", t.hour(), t.minute(), t.second());
    Serial.print(sz);
  }
  printInt(d.age(), d.isValid(), 5);
  smartDelay(0);
}
static void printStr(const char *str, int len)
{
  int slen = strlen(str);
  for (int i=0; i<len; ++i)
    Serial.print(i<slen ? str[i] : ' ');
  smartDelay(0);
}
```