

УДК 004.031

М.М. Поліщук, П.В. Саварин, Т.С. Олещук, Б.І. Марчук
Луцький національний технічний університет

РОЗРОБКА ПРОГРАМНО-АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ СТВОРЕННЯ GSM GPS- ТРЕКЕРА

М.М. Поліщук, П.В. Саварин, Т.С. Олещук, Б.І. Марчук. Розробка програмно-апаратного забезпечення для створення GSM GPS- трекера. В статті розглянуті системи передавання даних, проведено аналіз систем спутникового зв'язку. Описано етапи розробки програмно-апаратного забезпечення для GSM GPS-трекера на базі мікроконтролера STM32F407VET6. В статті описано вибір та перелік апаратного забезпечення розробленої системи передавання даних, а також прошивка мікроконтролера з під'єднанням до дисплея.

Ключові слова: модуля позиціонування, модуль GSM, дисплей; мікроконтролер, система зв'язку, конвертер, трекет, автономна система.

Н.Н. Полищук, П.В. Саварин, Т.С. Олещук, Б.И. Марчук. Разработка программно-апаратного обеспечения для создания GSM GPS- трекера. В статье рассмотрены системы передачи данных, проведен анализ систем спутниковой связи. Описаны этапы разработки программно-апаратного обеспечения для GSM GPS-трекера на базе микроконтроллера STM32F407VET6. В статье описано выбор и перечень аппаратного обеспечения разработанной системы передачи данных, а также прошивка микроконтроллера с подключением к дисплею.

Ключевые слова: модуля позиционирования, модуль GSM, дисплей микроконтроллер, система связи, конвертер, трекет, автономная система.

M. M Polishchuk, P.V. Savaryn, T.S. Oleschuk, B.I. Marchuk Development of software and hardware for creating GSM GPS tracker. In the article the systems of data transmission are considered, the analysis of satellite communication systems is carried out. The stages of development of hardware and software for the GSM GPS tracker based on the STM32F407VET6 microcontroller are described. The article describes the choice and list of hardware of the developed data transmission system, as well as firmware of the microcontroller with connection to the display.

Keywords: positioning module, GSM module, display; microcontroller, communication system, converter, track, autonomous system.

Постановка проблеми: Багато хто сьогодні використовує системи передавання даних під час подорожей, аби не заблукати та зрештою дістатися пункту призначення. У всьому світі стрімко зростає потреба передачі географічних координат на відстані. Для водіїв системи передавання даних, так звані GPS-навігатори, стали зручним провідником на місцевості, замінивши традиційні паперові мапи. Та сьогодні система глобального позиціонування використовується не лише для орієнтації у просторі, але й у багатьох інших, часом неочікуваних сферах життя.

Аналіз останніх досліджень і публікацій: З урахуванням аналізу різних літературних джерел [1-5]: система спутникового зв'язку – це створюване з використанням ракетно-стартових комплексів орбітальне угруповання космічних апаратів, допоміжних підсистем багаторівневої сукупності ліній зв'язку, що включає земні станції, середовище поширення, супутникові і земні ретранслятори, синтезуючі за певними умовами, принципами і критеріями у вигляді розподіленої функціональної структури з безліччю параметрів.

Супутниковий зв'язок є швидко розвиненим перспективним видом зв'язку, що обумовлено головним чином наступними його перевагами:

- можливість обслуговування абонентів, віддалених на значні відстані і розташованих в будь-яких регіонах Землі;
- простота реконфігурації систем спутникового зв'язку при зміні місць розташування абонентів;
- незалежність витрат при організації зв'язку від відстані між об'єктами;
- деякі впливи атмосферних явищ і географічних особливостей місць використання земної станції (супутникового телефону) на стійкість зв'язку.

Склад системи спутникового зв'язку, що має підмножини компонентів на різних рівнях, в самій загальній конфігурації представлено на рис. 1.

До складу систем спутникового зв'язку зазвичай включають:

- космічний сегмент (супутники, ретранслятори, антенно-фідерні пристрої);
- земний сегмент;
- ракетно-космічний комплекс;
- стартовий комплекс;
- підсистема управління.

Регламент радіозв'язку визначає «космічну систему» як «будь-яку групу, яка є спільно із земними і (або) космічними станціями, які використовують космічний зв'язок для певних цілей», а «супутникову систему» як «космічну систему, що використовує один або кілька штучних супутників землі».

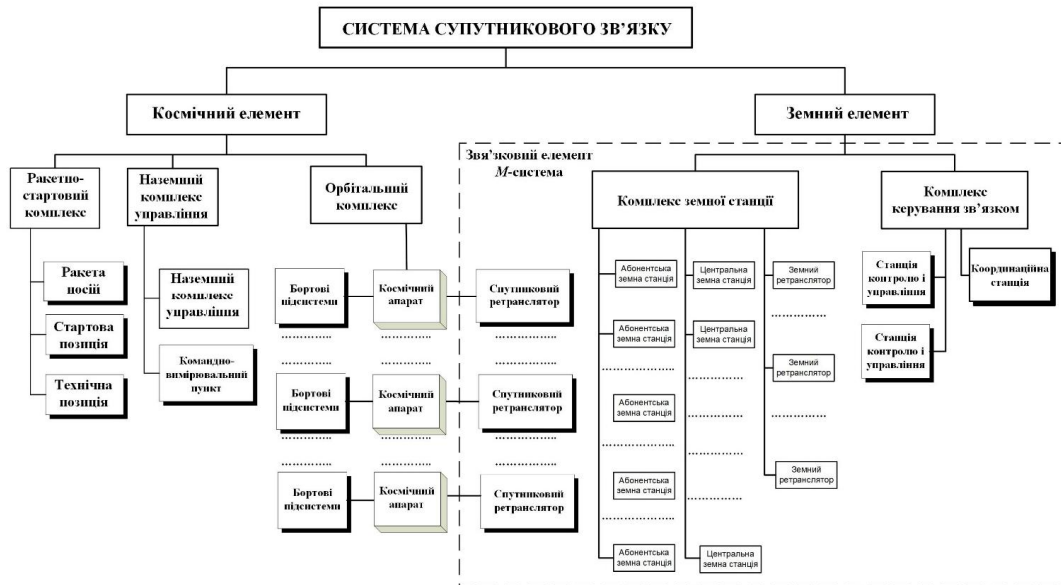


Рис. 1. Загальна конфігурація супутникової системи зв'язку

З урахуванням аналізу різних літературних джерел визначення систем супутникового зв'язку може бути сформульовано таким чином: система супутникового зв'язку – це створюване з використанням ракетно-стартових комплексів орбітальне угруповання космічних апаратів, допоміжних підсистем багаторівневої сукупності ліній зв'язку, що включає земні станції, середовище поширення, супутникові і земні ретранслятори, синтезуючі за певними умовами, принципами і критеріями у вигляді розподіленої функціональної структури з безліччю параметрів.

Рішення завдання синтезу багато-супутникової системи (часто називають М-системою) в найбільш загальному і закінченому вигляді досить складно, і мало можливо аналітичним шляхом. Зазвичай на практиці вибір структури, параметрів і характеристик систем супутникового зв'язку (орбітальних, енергетичних, частотних, сигнальних і пр.) залежить від багатьох умов і визначається різними міркуваннями. При цьому пріоритет часто віддається апробованим, традиційним технічним ідеям і рішенням, які мають аналоги [6].

Мета даної роботи полягає у створенні системи передачі даних GSM GPS-трекера на базі мікроконтролера STM32F407.

Виклад основного матеріалу. Супутникові навігаційні системи засновані на використанні координовані за рухом і випромінюванням сигналів мережі навігаційних штучних супутників Землі. Супутникові навігаційні системи забезпечують безперервне і практично миттєве визначення місця розташування і швидкості споживача в переважній більшості районів земної кулі (глобальні системи) з точністю, яка зазвичай значно перевищує точність інших навігаційних систем.

Структурна схема, створеної, автономної системи представлено на рис. 2, яка розроблена з використанням:

- модуля позиціонування на базі U-Blox NEO-7M,
- модуля GSM SIM800 MicroSIM,
- TFT LCD дисплей;
- мікроконтролера STM32F407.

Мікроконтролери STM32 здобувають все більшу популярність завдяки своїй потужності, досить різномірній периферії, та своєї гнучкості. У мікроконтролері STM32F407VET6 використовується тестова плата.

Цей мікроконтролер не є найпотужнішим серед STM32, але і не самий слабкий. Існують різні тестові плати, у томі числі і дуже дорогі, які коштують близько 20\$. На таких платах є майже те саме, що і на нашій платі, плюс програматор. В нашому випадку ми використовували програматор окремо.

Характеристики мікроконтролера STM32F407VET6

- ядро: 32-бітний RISC Corex-M4

- особливості: однократні інструкції DSP
- робоча частота: 168 МГц (210DMIPS / 1025 DMIPS / MHZ)
- робоча напруга: 1,8-3,6 В

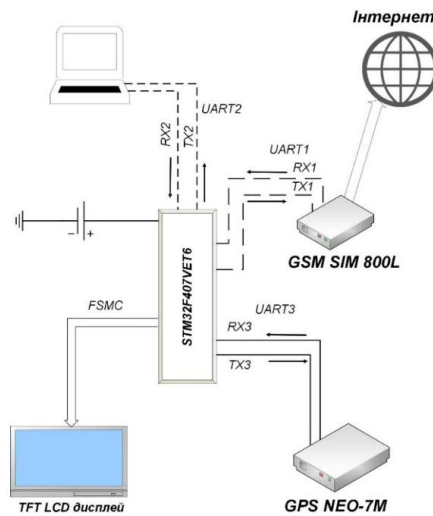


Рис. 2. Структурна схема GSM GPS-трекера

Будова плати мікроконтролера STM32F407VET6 представлено на рис 3.

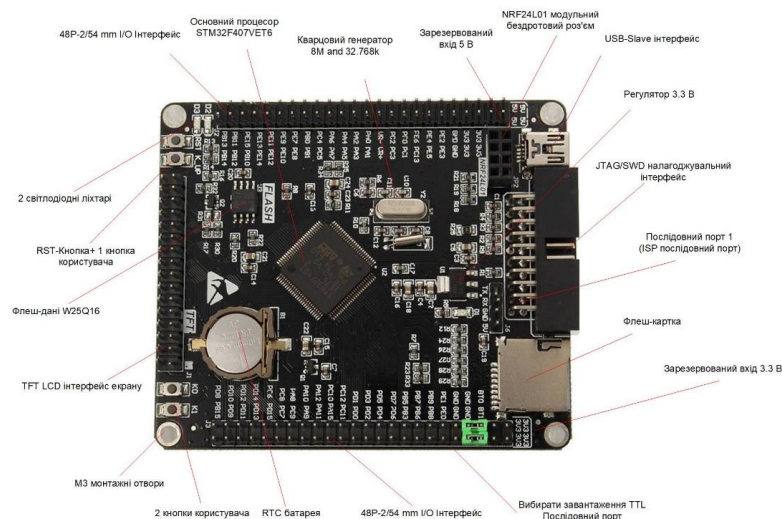


Рис. 3. Будова відлагоджувальної плати

Прошивка STM32 відбувалась за допомогою ST-Link програматора під Windows. При використанні програматора ST-Link піни BOOT0 і BOOT1 не використовуються і мають стояти у стандартному положенні для звичайної роботи контролера. Качаємо з сайту "st.com" утиліту "STM32 ST-LINK Utility". Встановлюємо її. З нею має бути встановлений і драйвер для ST-Link. Якщо ні, качаємо і встановлюємо драйвери ST-Link. Підключаємо ST-Link у USB- конектор, а відповідні виводи програматора підключаємо до виводів тестової плати згідно маркування.

Після завершення прошивки і перевірки завантажена програма мікроконтролера автоматично запуститься [7], як показано на рис. 4.

Підключення модуля GPS NEO-7M

GPS приймач на модулі uBlox NEO-7M забезпечує високу точність і стабільність частоти вихідного опорного сигналу. При цьому вартість плати становить всього 22-26 USD на eBay, 4\$ на Aliexpress. І що найцікавіше - вихідний опорний сигнал міг програмно встановлюватися від 0.25Гц до 10 МГц (на відміну від старого приймача Rockwell / Conexant / Navman Jupiter GPS, який мав вихід опорного сигналу тільки на 10 кГц) [8].

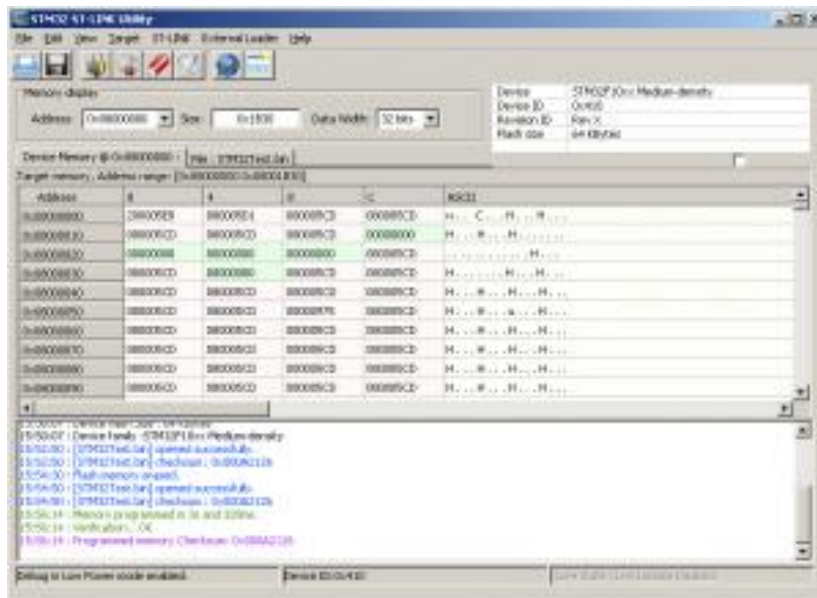


Рис. 4. Запуск програми мікроконтролера

Основні переваги GPS модуля від швейцарської компанії **u-blox** це – наявність GLONASS, хороша доступна документація і наявність власного сервісу AGPS.

Підтримка активних і пасивних антен, саме у 7M немає можливості програмно вимкати живлення активної антени, тому для енергозбереження потрібно зробити одну із запропонованих в документації схем апаратного виключення ззовні.

Інтерфейси: UART, SPI, I2C, USB

UART і I2C працюють одночасно, а ось SPI потрібно працювати за допомогою виведення D_SEL. Відповідно можна зробити кероване перемикання, наприклад підключивши GPIO до виходу D_SEL. Для наших цілей UART – це достатньо та дуже зручно. Також перемикання протоколу обміну немає великого сенсу, загальний вигляд модуля. Елементна база, яку я використовується цілком доступна в Україні. За винятком RF індуктивності 27nH, яку рекомендує виробник, схема представлена на рис. 5.

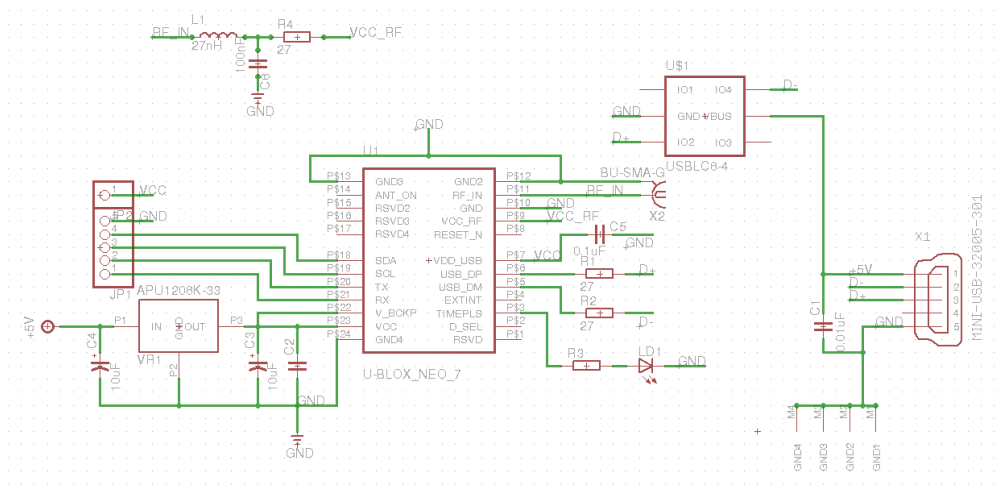


Рис. 5. Схема підключення модуля

На нижньому боці плати є 5 виходів:

- 2 виведення UART помічені як «Rx» «Tx»
- 2 виведення I2C помічені як «SDA» «SCL»
- 1 контакт для сигнальної «землі»

Також є SMA – female роз'єм для підключення активної / пасивної антени. На верхній стороні є самоточний пін 3.3В. Він не був запланований і не був розлучений на PCB спочатку, звідси вийшов висить в повітрі конденсатор, антена використовується активна.

У u-blox є свій власний БНАР протокол, який нарівні з NMEA вміє передавати інформацію про місцезнаходження, і багато іншого. Всі елементи працюють у зв'язці з PC і Cubieboard.

Основний недолік цього модуля полягає в тому, що GPS і GLONASS не можуть працювати одночасно. Доводиться програмно перемикатися між ними [9].

Підключення TFT LCD дисплея до мікроконтролера. Використання FSMC

Однією особливістю мікроконтролерів STM32 – а саме модуль FSMC, що є практично незамінною при роботі із зовнішньою пам'яттю, або, наприклад, з графічним дисплеєм. Власне потрібно підключити дисплей за допомогою модуля FSMC.

FSMC реалізує паралельний інтерфейс обміну даними між різними пристроями, це просто паралельна шина. Використовуючи FSMC при роботі із зовнішньою пам'яттю, ми отримуємо можливість включити зовнішню пам'ять в адресний простір мікроконтролера. Це дає звернення до зовнішньої пам'яті, що значно спрощується – необхідно просто звертатися до ОЗУ мікроконтролера за певними заданими адресами. Тобто всю роботу з тимчасовими діаграмами, таймінгами і іншим модуль FSMC бере на себе. Потрібно просто написати дані за адресою, а FSMC смикає лінії даних, повністю здійснюючи безпосередню роботу з тим, що підключається пристроєм.

Схожим чином працює вся ця справа і при підключенні дисплеїв. Але все-таки тут трохи інакше. Нехай у нас є дисплей, у якого є такі виходи: DB [17: 0] - 18 ліній для передачі даних (не забуваємо, що тут у нас паралельна передача даних, а не послідовна)

Також є виходи для вирішення запису/читання – туди ми повинні видавати синхро-імпульси в певній послідовності. Крім того, у дисплея є виходи: chip select'a, reset і т.д.

Щоб всім цим управляти, нам і допоможе FSMC. І не просто допоможе, а всю роботу візьме на себе. Якщо вже підключений дисплей правильно, написана програма для FSMC STM32. Робота дисплея схожа на роботу із зовнішньою пам'яттю. Потрібно лише вносити дані за певною адресою, а FSMC буде в цей час розподіляти їх лініями даних, стежити за часовими інтервалами, передавати всі інші необхідні дисплею сигнали. А щоб розділити передачу даних і команд ми повинні записувати байти за різними адресами.

У дисплея є вихід для вибору – дані/команда. Тобто станом цього виходу дисплей вирішує, що саме зараз до нього надійде. А у FSMC є шина адреси і шина даних. Так ось цей вихід дисплея підключається до якогось піну шини адреси. І якщо підключити його до 16 виходу шини адреси, то записавши який-небудь байт з будь-якого (!) Адресу з нульовим 16 бітом ми подамо дисплею команду.

Коли вирішили підключений 16 біт шини адреси, тоді беремо адресу з доступних FSMC - 0x60000000. Бачимо, що в цьому значенні 16 біт дорівнює нулю, а значить, коли ми запишемо значення за цією адресою, FSMC подасть дисплею сигнал на запис, також повідомить дисплею, що зараз буде команда, ну і, звичайно ж, видасть на шину самі дані. А якщо ми запишемо значення за адресою 0x60010000 (16 біт - одиниця) То FSMC все вирішить і передасть дисплею дані. Ось так все просто). Але, якщо, у 16-бітному режимі роботи FSMC 16 біт шини адреси відповідає 17-му бітку адреси (тобто 0x60020000). Тоді, всі інші адреси також будуть зміщені на 1 біт в цьому режимі.

Робота з дисплеєм і з зовнішньою пам'яттю з точки зору програміста виглядає однаково, але на ділі все не так. При роботі з дисплеєм ніяка пам'ять нікуди не «проектується», умовно пишуться дані за адресами, але це всього лише дає FSMC сигнал про те, що пора починати діяти

Для роботи з FSMC було використано Standard Peripheral Library. Там все аналогічно будь-який інший периферії, хіба що налаштувань побільше.

Робота з FSMC буде за допомогою плати MiniSTM32 [10]. Там уже встановлений дисплей, так що ніяких зайвих рухів тіла не буде потрібно, схема підключення дисплея представлено на рис. 6

Якщо звернути увагу на терморегулятори конкретного дисплея і на виходи FSMC мікроконтролера, то видно, що вони точно відповідають один одному.

Для того щоб фарбувати дисплей в певний колір потрібно спочатку залити екран червоним, потім зеленим і потім синім (RGB). Отже, потрібно отримати миготливий дисплей, на якому змінюють один одного три кольори. Дані будемо передавати в 16-бітному режимі (рис. 7).

Червоному кольору відповідає - 111111 000000 000000.

Зеленому - 000000 111111 000000.

Синього - 000000 000000 111111.

Як видно з рис. 7, на 18 біт кольору припадають 16 біт даних, в результаті отримуємо наступне:

- для червоного – 0xF800;
- для зеленого – 0x07E0;
- для синього – 0x001F.

Слідуючий етап є написання програми. Ініціалізація дисплея взята з китайських прикладів програм, які йшли разом з платою.

Число 76800 – кількість точок дисплея. Все інше начебто зрозуміло, з настройками FSMC теж ясно, як завжди в SPL все поля структури названі логічно і адекватно їх функції. Після прошивки програми в контролер дисплей починає справно підморгувати [11].

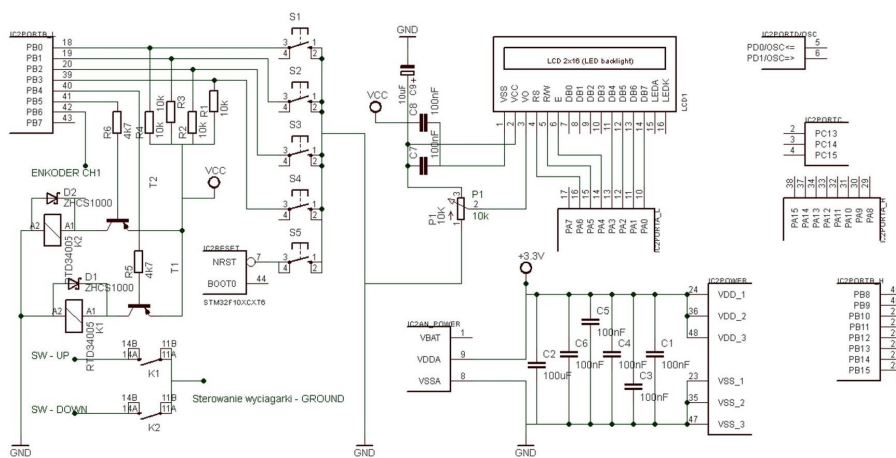


Рис. 6. Схема підключення дисплея

80-system 16-bit interface

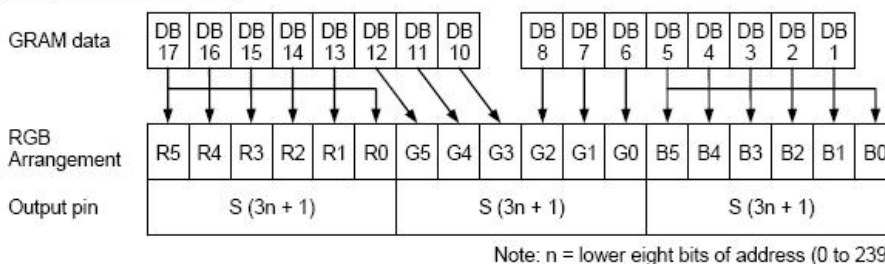


Рис. 7. 16-ти бітний режим передавання

Для перевірки працездатності пристрою досить мати ПК і сімкарту з невеликим грошовим рахунком. Перевірку працездатності можна проводити за двома методами підключення модуля до ПК: через перетворювач інтерфейсів USB-UART або використовуючи Arduino UNO. Розглянемо метод перевірки без Arduino. Встановлюємо сімкарту в модуль GSM GPRS, дотримуючись розташування контактів. З'єднайте ПК через перетворювач інтерфейсів USB-UART з пристроєм за схемою наведеної вище. Підключіть живлення номінальною напругою. Дочекайтеся орієнтуючись на мигання світлодіода підключення до мобільного оператора. Увімкніть на ПК термінальну програму. Дані в неї слід вводити великими буквами. Використовуючи термінальну програму відправляємо в модуль GSM GPRS SIM800 через порт підключення наступні команди.

AT

Відповідь модуля ОК

AT + CSQ

Відповідь модуля + CSQ: 18,0 ОК

Ця команда дає інформацію про рівень сигналу. Перше число – рівень сигналу, величина 18 означає – 78 dbm. Друге число – кількість помилково прийнятих біт, величина "0" говорить про частку помилок менше 0,2%, що свідчить про хорошу зв'язку (рис. 8).

Використовуємо перетворювача USB-UART, то можна використовувати Arduino UNO. Потрібно з'єднати контакт RX модуля SIM800L з контактом 11 цифрового сигналу Arduino UNO застосовуючи резисторний дільник напруги. Після цього з'єднуємо контакт TX модуля SIM800L з контактом 10 цифрового сигналу Arduino UNO. Завантажимо в Ардуіно програму, текст якої наведено далі. Тут використовується бібліотека SoftwareSerial дозволяє реалізувати послідовний інтерфейс на будь-яких цифрових висновках Ардуіно за допомогою програмних засобів, які дублюють функціональність UART [12].



Рис. 8. Відправка команд у модуль

Текст програми:

```
#include
SoftwareSerial mySerial(10, 11); // RX, TX
void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  mySerial.begin(9600);
}
void loop() // run over and over
{
  if (mySerial.available())
    Serial.write(mySerial.read());
  if (Serial.available())
  {
    while(Serial.available())
    {
      mySerial.write(Serial.read());
    }
    mySerial.println();
  }
}
```

Підключення конвертера CH340G CH340 модуля USB к TTL

Для USB-моста CH340G на комп'ютері необхідно встановити драйвер, остання офіційна версія якого знаходиться на сайті виробника. Якщо вам не вдається розібратися в китайській грамоті, то копії драйверів можна завантажити звідси. Після звичайної установки драйвера необхідно переконаватися, що при включенні модуля в USB-роз'єм з'являється новий COM-порт «USB-SERIAL_CH340» і подивитися який номер порту система виділила для нього.

Узагальнений шлях до портів в Win-операціонках: *Панель управління / Система / Обладнання / Диспетчер пристроїв / Порти (COM)*. якщо далі пройти по *Властивості / Параметри / Додатково*, то можна змінити номер порту і швидкість роботи за замовчуванням. Однак слід пам'ятати, що зазвичай швидкістю порту управляє прикладна програма.

Для перевірки працездатності модуля треба з'єднати між собою лінії RXD і TXD і відсилаючи будь-яку термінальну програмою символи, переконається в їх поверненні підключення пристрою якщо мікроконтролер має своє харчування, то модуль підключається трьома проводами: GND, TxD, RxD.

Приймач модуля RxD треба підключати до передавача мікроконтролера TX і, відповідно, TxD до RX! Рівень сигналів на лініях TxD, RxD задається джампером, який з'єднує лінію VCC з висновками 3.3V або 5.0V. щоб виключити перекося між модулем і мікро контролером при виключенні живлення, рекомендується на час налагодження з цих же висновків взяти харчування для плати мікроконтролера – до 400mA від виведення 5.0V і до 50mA від 3.3V. В цьому випадку з'єднувальний джгут має 4 дроти і перемичку замість джампера [13]. Схема підключення до мікроконтролера представлено на рис. 9.

Розроблена автономна система передачі даних, яку можна застосувати для отримання координат місця знаходження автомобілі або іншого транспортного засобу. Прилад є досить надійним, здатний безвідмовно виконувати задані функції протягом тривалого часу. Плата має невеликі розміри, споживає малу кількість електроенергії.

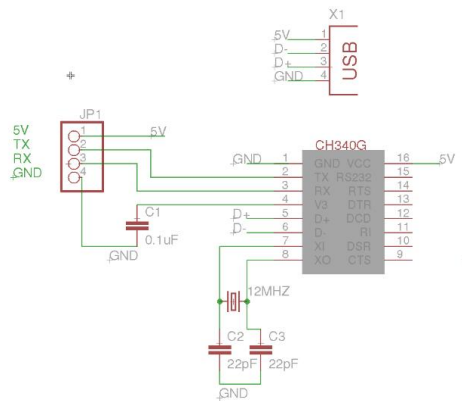


Рис. 9. Схема підключення конвертера

Загальний вигляд автономної системи представлено на рис. 10.

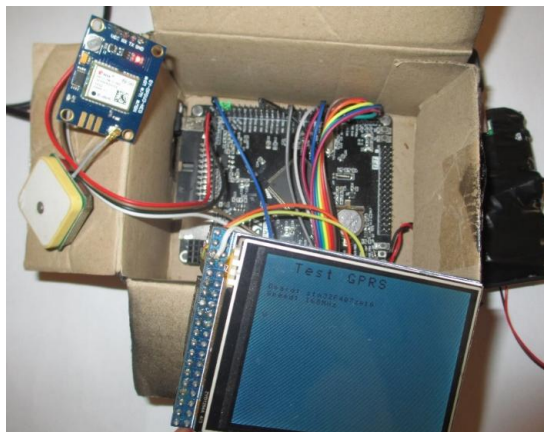


Рис. 10. Фото розробленого GSM GPS-трекер на базі мікроконтролера STM32F407

1. Передавання даних [Електронний ресурс] – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Передавання_даних – Назва з екрану (Дата звернення 22.11.16)
2. Синхронна передача [Електронний ресурс] – Режим доступу до ресурсу: http://contes.te.ua/sinkhronnaja_peredacha.html – Назва з екрану (Дата звернення 10.09.2016)
3. Антонюк Анатолій Олександрович. Основи захисту інформації в автоматизованих системах: Навч. посіб. / Національний ун-т "Києво-Могилянська академія". - К. : КМ Академія, 2003. - 244с.
4. Білинський Й.Й. Типова система передачі даних / Й.Й. Білинський, К.В. Огородник, М.Й. Юкиш // Електронні системи (посібник) [Електронний ресурс] – Режим доступу до ресурсу: http://posibnyky.vntu.edu.ua/e_s/5.htm – Назва з екрану (Дата звернення 27.11.2016)
5. Типова система передачі даних [Електронний ресурс] – Режим доступу до ресурсу: http://studopedia.su/5_16071_tipova-sistema-peredachi-danih.html – Назва з екрану (Дата звернення 01.12.2016)
6. Андрій Яшук. Системи безпроводних технологій передачі даних: Навчальний посібник / Матеріали були підготовлені в рамках Проекту ПЛ-НТУ. Транскордонний обмін досвідом ІРВU.03.01.00-06-386/11-00, - 313 с.
7. STM32. Програмування STM32F103. Тестова плата. Прошивка через послідовний порт та через ST-Link програматор [Електронний ресурс] – Режим доступу до ресурсу: http://www.avislab.com/blog/stm32_st_link/ – Назва з екрану (Дата звернення 31.01.2018).
8. uBlox NEO-7M: GPSDO или путь к GPSDO? [Електронний ресурс] – Режим доступу до ресурсу: <http://www.ra3arpw.ru/proekty/ublox-neo-7m/> – Назва з екрану (Дата звернення 04.03.2018).
9. Модуль позиціонування на базі u-blox NEO-7M [Електронний ресурс] – Режим доступу до ресурсу: <https://habrahabr.ru/post/233925/> – Назва з екрану (Дата звернення 15.03.2018).
10. Быстрый старт с отладочной платой Mini STM32 [Електронний ресурс] – Режим доступу до ресурсу: <http://microtechnics.ru/stm32-i-displej-ispolzovanie-fsmc/> – Назва з екрану (Дата звернення 10.03.2018).
11. STM32 и дисплей. Использование FSMC. [Електронний ресурс] – Режим доступу до ресурсу: <http://microtechnics.ru/bystryj-start-s-ofladочноj-platoj-mini-stm32/> – Назва з екрану (Дата звернення 11.03.2018).
12. Модуль GSM GPRS SIM800 MicroSIM с антенной и ардуино [Електронний ресурс] – Режим доступу до ресурсу: <http://www.electronic52.in.ua/proekty-arduino/modul-gsm-gprs-sim800-microsim-s-antenoj> – Назва з екрану (Дата звернення 04.04.2018).
13. CH340G CH340 модуля USB к TTL [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.aliexpress.com/item/Free-Shipping-CH340-module-USB-to-TTL-CH340G-upgrade-download-a-small-wire-brush-plate-STC/32713914780.html?spm=2114.13010608.0.0.8Oq4Oj> – Назва з екрану (Дата звернення 14.01.2018).