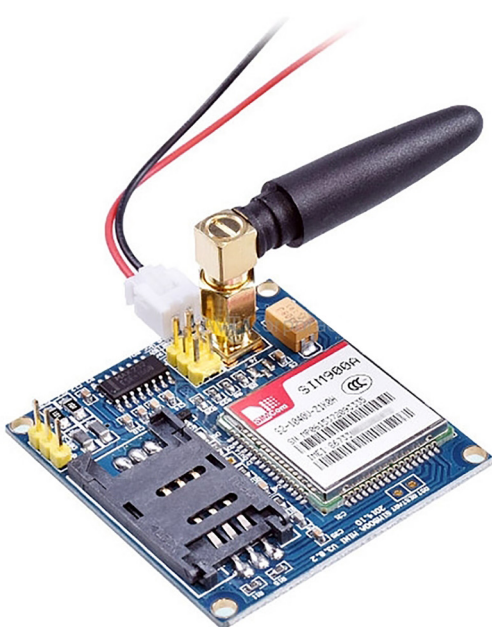


Restarter do routera sterowany za pomocą SMS

Nikogo nie trzeba przekonywać o tym, jak bardzo jesteśmy uzależnieni od Internetu. Nie trzeba też przekonywać, jak irytujące mogą być niespodziewane przerwy w dostępie. Wiele z takich awarii można łatwo usunąć poprzez zrestartowanie routera lub modemu. Oczywiście, o ile mamy do niego łatwy dostęp. Jeśli nie, to idealnym rozwiązaniem będzie prezentowane urządzenie. Może służyć do zdalnego włączania, wyłączenia lub restartowania takich urządzeń jak np. routery. Jedyne, co musimy zrobić, to wysłać SMS.



Fotografia 1. Moduł z układem SIM900A

Głównym elementem urządzenia jest płytko Arduino Mega. Wersja „Mega” ma tę ważną cechę, że wyposażono ją w 4 sprzętowe interfejsy szeregowy UART (wersja Uno ma tylko 1). Pierwszy interfejs zarezerwowano dla komunikacji poprzez USB, dzięki czemu jest możliwe śledzenie pracy układu oraz programowanie płytki Arduino. Drugi UART służy do komunikacji z modułem GSM, pozostałe są nieużywane.

Do płytki Arduino dołączono moduły rozszerzające AVT1633 oraz AVT1615 udostępniające szereg złączy śrubowych na wyprowadzeniach Arduino oraz nieskomplikowany interfejs użytkownika w postaci wyświetlacza 2×16 i 4 przycisków.

Drugim ważnym elementem urządzenia jest moduł GSM. Na płytce znajduje się układ SIM900A oraz peryferia niezbędne do jego prawidłowej pracy. Należy tylko dołączyć zasilanie oraz interfejs szeregowy UART, jak pokazano na fotografii 1.

Pozostałe elementy urządzenia to regulowany stabilizator napięcia AVT1066 z dodatkowym zasilaczem 12 V oraz moduł wykonawczy z przełącznikiem AVT1656 z małą modyfikacją – wykorzystuje styki NO przełącznika a nie NC, dzięki czemu sterowany

REKLAMA

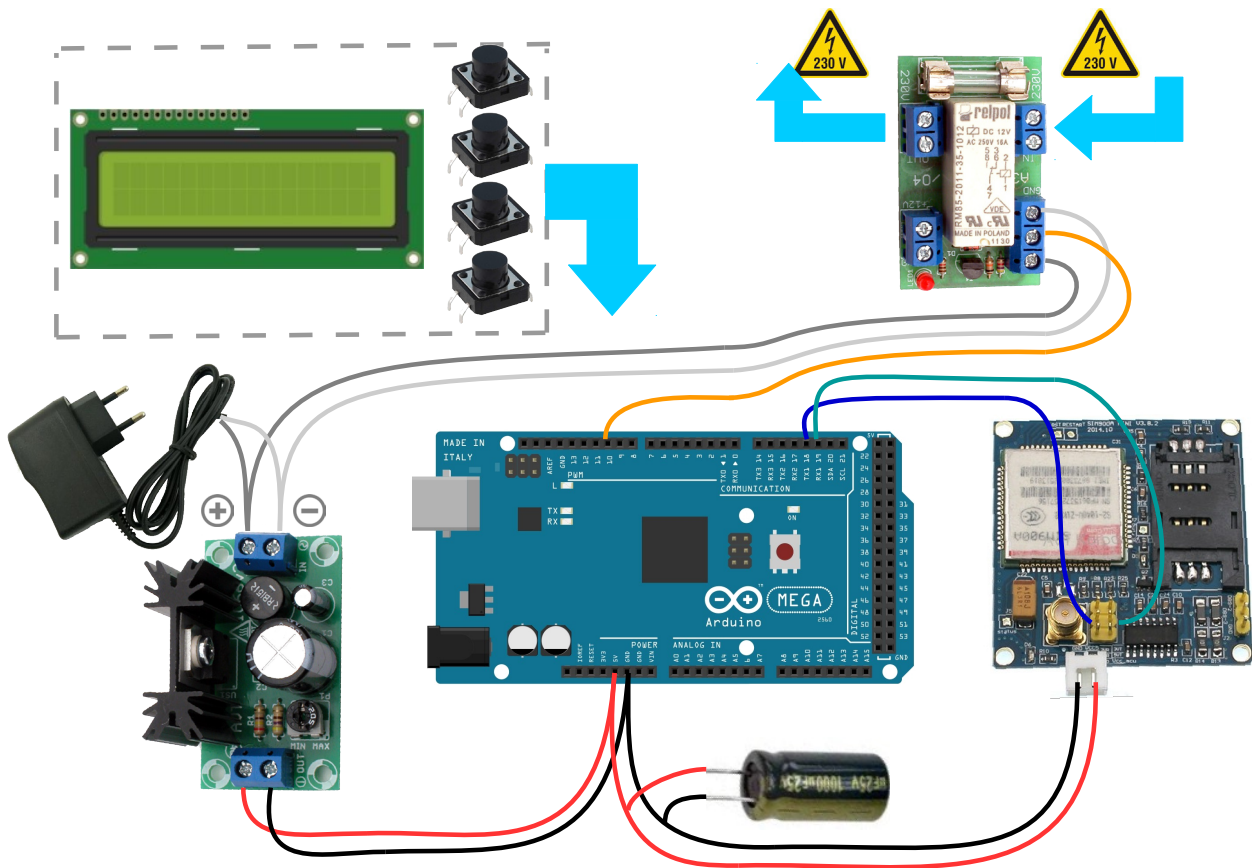
Projekty na

STM32

www.stm32.eu

ST life.augmented

KAMAMI



Rysunek 2. Schemat ideowy połączeń

obwód jest załączony nawet gdy urządzenie nie działa.

SIM900 a SIM900A

W zastosowanym module GSM zamontowany jest układ SIM900A – jedna z wersji popularnego układu SIM900. Różnica pozornie niewielka, ale można stracić wiele godzin, próbując zmusić wersję z literką „A” do załogowania się do naszych sieci GSM. Jest to wersja przygotowana na rynek azjatycki i przystosowana do tamtejszych sieci. Aby układ działał z sieciami europejskich operatorów, należy zmienić oprogramowanie układu, tzw. firmware.

Do przeprowadzenia tej operacji potrzebne będą: aplikacja do zmiany firmware’u układu SIM900, plik firmware’u, konwerter USB-UART np. AVTMOD09 oraz źródło zasilania dla modułu. Aplikację można pobrać ze strony <http://goo.gl/JaG5iZ>. Zbiór różnych wersji firmware do układów SIM900 znajduje się na tej stronie <http://goo.gl/0nMQEZ>. Należy wybrać najnowszą wersję dla układu SIM900 (bez żadnej literki na końcu), na tę chwilę jest to: 137B15SIM900M64_ST.

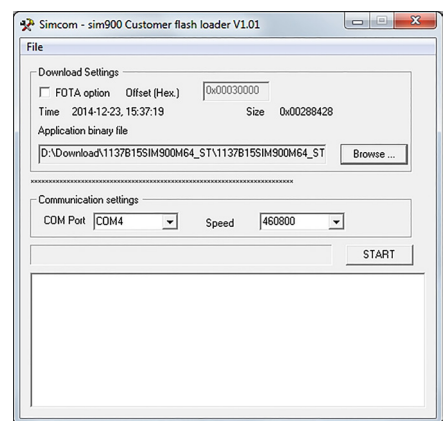
Po pierwsze należy połączyć konwerter do komputera i sprawdzić, który port COM został mu przydzielony. Następnie należy połączyć konwerter z modulem (RX, TX oraz GND zgodnie z oznaczeniami) oraz przygotować źródło

zasilania, jak na **rysunku 2**. W tym przypadku jest to regulowany stabilizator AVT1066 z ustawionym napięciem na wyjściu 5,0 V. Na razie nie należy dołączać zasilania do modułu. W drugim etapie należy uruchomić pobraną aplikację *Simcom – sim900 Customer flash loader V1.01.exe*. Zostanie wyświetlone okno, jak na **rysunku 3**. Należy wskazać plik firmware z rozszerzeniem *.cla*, wskazać port COM konwertera USB-UART i kliknąć przycisk *START*. Teraz należy dołączyć zasilanie modułu, rozpocznie się wtedy aktualizacja, może potrwać kilka-kilkanaście minut. Po zakończeniu należy odłączyć na chwilę zasilanie modułu i gotowe.

Zasilanie SIM900

Bardzo ważnym czynnikiem, który wpływa na poprawną pracę modułu jest jego zasilanie. Sam układ SIM900 wymaga zasilania napięciem z przedziału 3,4...4,5 V, na płycie modułu dodatnia linia zasilania poprowadzona jest poprzez diodę prostowniczą na której następuje spadek napięcia ok 0,7 V, zatem moduł można bezpiecznie zasilac napięciem 5 V.

Moduł w stanie czuwania pobiera niewielki prąd, kilkanaście...kilkadziesiąt mA ale w trakcie komunikacji z siecią GSM może pobierać chwilowo nawet 2...3 A. Dlatego nie ma sensu zasilanie modułu ze złącza USB czy z płytki ARDUINO, czy nawet z ładowarki USB. Musi być



Rysunek 3. Aplikacja do zmiany firmware układu SIM900

zasilacz o napięciu 5 V i wydajności ok 1,5 A a dodatkowo, blisko modułu, na szynach zasilania musi być zamontowany kondensator typu *LOW ESR* (do pracy impulsowej) o pojemności min 1500 μ F. Prezentowane urządzenie było zasilane zasilaczem 12 V/1,5 A z dołączonym regulowanym stabilizatorem AVT1066 z ustawionym nap na wyjściu 5,0 V i dodatkowym kondensatorem 1500 μ F/6,3 V low esr.

Komunikacja z SIM900

Komunikacja z modulem odbywa się poprzez dwu-przewodowy interfejs UART (RX i TX). Domyślne ustawienia komunikacji to: 19200, 8, n, 1. Za pomocą konwertera USB-UART i programu typu terminal

Listing 1. Inicjowanie zasobów sprzętowych

```
void setup() {
//początkowy stan wyjścia - stan niski LOW
digitalWrite(output1_pin, LOW);
//konfigurowanie wyjść
pinMode(button1_pin, INPUT_PULLUP);
pinMode(button2_pin, INPUT_PULLUP);
pinMode(button3_pin, INPUT_PULLUP);
pinMode(button4_pin, INPUT_PULLUP);
pinMode(output1_pin, OUTPUT);
//konfigurowanie wyświetlacza
lcd.begin(16,2);
lcd.print(„Hello”);
//konfigurowanie interfejsu szeregowego UART1
//do komunikacji z modułem GSM
Serial.begin(19200);
Serial1.begin(19200);
//konfigurowanie interfejsu szeregowego UART0
//do komunikacji przez USB
Serial.println(„Hello”);
...
}
```

Listing 2. Konfigurowanie modułu GSM

```
//wybór komendy dla modułu GSM
switch (gsm_step) {
case 0: {
//wyłącza echo
Serial1.print(„ATE0\r”);
} break;
case 1: {
//zezwała na logowanie do sieci GSM
Serial1.print(„AT+CREG=1\r”);
} break;
case 2: {
//sprawdza stan zalogowania do sieci GSM
Serial1.print(„AT+CREG?\r”);
} break;
case 3: {
//pobiera z SIM nazwę operatora
Serial1.print(„AT+COPS?\r”);
} break;
case 4: {
//określa tekstowy tryb odczytu SMS
Serial1.print(„AT+CMGF=1\r”);
} break;
...
}
```

Listing 3. Sprawdzanie poziomu sygnału i odczyt wiadomości tekstowej z pozycji 1

```
//wybór komendy dla modułu GSM
switch (gsm_step) {
...
case 5: {
//sprawdza siłę sygnału sieci GSM
Serial1.print(„AT+CSQ\r”);
} break;
case 6: {
//odczytuje SMS z pozycji 1
Serial1.print(„AT+CMGR=1\r”);
} break;
...
}
```

Listing 4. Sprawdzanie komend zawartych w wiadomości tekstowej

```
//jeśli odczytano SMS
if (command == „CMGR”) {
...
//jeśli SMS zawiera treść
if (StringSmsGet() > 0) {
...
//jesli polecenie załączenia
if (string_sms == „Out on”) {
digitalWrite(output1_pin, LOW);
lcd.setCursor(0, 0);
lcd.print(„ OUTPUT ON ”);
autorefresh = 0;
//następny etap - wysyłanie potw.
gsm_step = 7;
}
//jesli polecenie wyłączenia
if (string_sms == „Out off”) {
digitalWrite(output1_pin, HIGH);
lcd.setCursor(0, 0);
lcd.print(„ OUTPUT OFF ”);
autorefresh = 0;
//następny etap - wysyłanie potw.
gsm_step = 7;
}
//jesli polecenie restartu
if (string_sms == „Out res”) {
digitalWrite(output1_pin, HIGH);
lcd.setCursor(0, 0);
lcd.print(„ OUTPUT RESTART ”);
delay(2000);
digitalWrite(output1_pin, LOW);
autorefresh = 0;
//następny etap - wysyłanie potw.
gsm_step = 7;
}
}
//jesli nie ma SMS
}
```

Listing 5. Wysyłanie wiadomości potwierdzającej

```
//wybór komendy dla modułu GSM
switch (gsm_step) {
...
case 7: {
//odpisuje sms na numer który przysłał wiadomość
Serial1.print(„AT+CMGS=\”);
Serial1.print(string_number);
Serial1.print(„\r”);
delay(300);
Serial1.print(„OK”);
Serial1.write(0x1A);
} break;
}
```

można w sterować modułem GSM. Sterowanie odbywa się komendami AT. Aby przetestować komunikację można wysłać komendę AT<CR> (symbol <CR> oznacza znak końca linii, zwykle odpowiada mu przyciśnięcie klawisza Enter). Moduł w odpowiedzi wyśle <CR><LF>OK<CR><LF>. Symbol <LF> oznacza początek nowej linii). Treść odpowiedzi modułu zwykle ma właśnie taką strukturę: <CR><LF>odpowieź modułu<CR><LF>. W dalszej części artykułu, dla lepszej czytelności pomijane będą <CR> i <LF>. Inną pomocną komendą jest AT+GSV w odpowiedzi moduł wyśle informacje o typie zastosowanego układu SIM900 oraz o wersji firmware.

Po zainstalowaniu karty sim, można sprawdzić czy moduł zalogował się do sieci GSM. Uwaga – kartę SIM należy instalować tylko przy wyłączonym zasilaniu modułu. Ponadto, karta musi być aktywna (aktywowana wcześniej w telefonie) i dla ułatwienia powinna mieć wyłączone żądanie kodu PIN. Najpierw należy wysłać komendę

AT + C R E G = 1 , a następnie, gdy po wysłaniu AT+CREG=? otrzymamy odpowiedź +CREG=1,1 to znaczy, że moduł jest zalogowany.

Działanie programu

Program wykonano za pomocą środowiska Arduino. Pierwsze zadania, które wykonuje, to inicjowanie zasobów sprzętowych – ustawienie kierunku wyprowadzeń, ruch wyświetlacza, uruchomienie portów szeregowych i na koniec wyświetlenie komunikatu Hello (listing 1). Następny etap polega na skonfigurowaniu modułu GSM, a gdy moduł zaloguje się do sieci, to z karty sim odczytywana zostaje nazwa operatora i wyświetlona na wyświetlaczu (listing 2). Potem program w pętli sprawdza na zmianę dwie rzeczy: poziom sygnału GSM czyli tzw. zasięg, który również jest prezentowany na wyświetlaczu oraz próbuje odczytać SMS (listing 3). Wiadomości tekstowe zapisywane są w pamięci karty sim,

pod na pierwszej wolnej pozycji. Program zawsze odczytuje wiadomość na pozycji 1, jeśli zawiera jakąś treść to sprawdza i ewentualnie wykonuje komendy a na koniec kasuje tę wiadomość. W ten sposób kolejna wiadomość zostanie zapisana na tej pozycji (listing 4).

Urządzenie akceptuje 3 komendy (smsy o takich treściach): Out off – powoduje wyłączenie wyjścia, Out on – powoduje załączenie wyjścia oraz Out res – powoduje rozłączenie wyjścia na ok 2 s a następnie automatyczne załączenie, czyli restart odbiornika dołączonego poprzez moduł wykonawczy. Jeżeli SMS zawiera poprawną komendę to urządzenie wykonuje zadanie a następnie wysyła potwierdzenie – smsa o treści OK, pod numer nadawcy (listing 5).

Pełne źródło programu jest dostępne w materiałach dodatkowych do projektu.