**Jak jednoduše na GSM –VII.díl – Knihovna pro Arduinotech GSM Shield**

Úvodní hrátky s GSM shieldem Arduinotech máme za sebou, někomu by se možná mohl zdát kód prezentovaný v minulém dílu trochu těžkopádný a je to pravda. Z Arduino světa jsme zvyklí mít všechno hezky dostupné z nějaké knihovny a ani náš vývojový tým nebude výjimkou a pro Arduinotech GSM shield jsme vypracovali knihovnu základních funkcí, které jsou potřeba pro zacházení s hovorem a SMSkou. Tato knihovna bude dále rozvíjená a na její aktualizace budete vždy na našich stránkách upozorněni včetně nějakých příkladů použití a popisů nových funkcí. V některých reakcích od Vás jsem zaznamenal, že by jste rádi nějaké příklady pro GPRS komunikaci – ano budou v rámci rozvoje knihovny již brzy.

Instalaci samotné knihovny nebudu popisovat, předpokládám již nějakou zručnost v tomto ohledu, nicméně návodu, jak to udělat je dost jak na zahraničních, tak českých webech.

**ArduinotechGSMShield knihovna**

Probereme jednotlivé veřejné procedury této knihovny, jejich vstupy a výstupy a použití budeme demonstrovat na jednoduchém sketchi, který čeká na zavolání a po vyhodnocení, zda je číslo oprávněno volat pak na 2 sekundy přitáhne relé 1. Tato mini aplikace se hodí např. pro ovládání vjezdové brány, kdy podobná zařízení se na netu prodávají za nehorázně vysoké ceny. S naším shieldem a Arduino UNO jste na 1/3 nákladů a navíc máte celé ovládání pod kontrolou, takže doplnit jej o ovládání SMSkou nebo další funkce nebude žádný problém. Pojďme ale postupně k jednotlivým funkcím:

**Třída AGS**

Název třídy: AGS (zkratka ArduinotechGSMShield)

Vstupní parametry: debug mód -> 1=zapnutý, 0=vypnutý

Výstupní parametry: žádné

Příklad:

vytvoření instance třídy ve sketchi:

**AGS modul(1); //debug mód povolen**

**AGS modul(0); //debug mód zakázán**

**begin();**

Popis: inicializuje shield, definuje softwareSerial na portech RX=2, TX=3, rychlost 9600 b/s, inicializuje sériový kanál Arduina na rychlosti 9600b/s, spouští konfigurační proces SIM800Init()

Vstupní parametry: žádné

Výstupní parametry: žádné

Příklad:

**modul.begin();**

**SIM800Init();**

Popis: Konfigurační procedura pro SIM800 modul, lze použít samostatně např. pro inicializaci v případě zjištění nekorektního chování v určitých pasážích kódu. Tato procedura je zahrnuta do funkce begin();

Vstupní parametry: žádné

Výstupní parametry: žádné

Příklad:

**modul.SIM800Init();**

**checkCallAndSMS();**

Popis: Provádí otestování, zda je přítomen příchozí hovor nebo nová SMSka. Doporučuji vložit do hlavní smyčky

Vstupní parametry: žádné

Výstupní parametry: 0 = žádný hovor nebo nová SMS, 1 = nový hovor, 2 = nová SMS

Příklad:

**loop()**

**{**

**infoStatus = modul.checkCallAndSMS();**

**if (infoStatus == 1)**

**{**

**//akce při detekci volání**

**}**

**if (infoStatus == 2)**

**{**

**//akce při detekci SMS**

**}**

**}**

**callEnd();**

Popis: Vyšle příkaz do SIM800 pro ukončení stávajícího hovoru – zavěšení

Vstupní parametry: žádné

Výstupní parametry: žádné

Příklad:

**modul.callEnd();**

**makeCall(String callNumber);**

Popis: Provede vytočení čísla – sestaví hlasový hovor

Vstupní parametry: String telefonní číslo v národním nebo mezinárodním tvaru bez +

Výstupní parametry: žádné

Příklad:

**modul.makeCall(“739822476“);**

**getNumber();**

Popis: Vrátí řetězec s telefonním číslem voljícího nebo účastníka, který poslal SMS. Obvykle je tento dotaz realizován bezprostředně po vyhodnocení stavu checkCallAndSMS() funkce.

Vstupní parametry: žádné

Výstupní parametry: String telefonní číslo

Příklad:

**infoStatus = modul.checkCallAndSMS();**

**if (infoStatus == 1) Serial.println(“Volajici ucastnik:” + modul.getNumber());**

**if (infoStatus == 2) Serial.println(“SMS od:” + modul.getNumber());**

**getSMSContent();**

Popis: Zjištění obsahu přijaté SMS

Vstupní parametry: žádné

Výstupní parametry: String obsah SMS

Příklad:

**if (infoStatus == 2)**

**{**

**Serial.println(“SMS od:” + modul.getNumber());**

**Serial.println(“Obsah SMS:” + modul.getSMSContent());**

**}**

**sendSMS(String number,String sms);**

Popis: funkce pro odeslání SMS na dané číslo s daným obsahem

Vstupní parametry: String číslo příjemce a String obsah SMS

Výstupní parametry: žádné

Příklad:

**modul.sendSMS(“73982246”,”Ahoj!”);**

**getProviderName();**

Popis: Vrací řetězec se jménem operátora GSM, které je prezentováno v rámci GSM sítě

Vstupní parametry: žádné

Výstupní parametry: String jméno operátora

Příklad:

**Serial.println(“GSM operator:” + getProviderName());**

**getQuality();**

Popis: Vrací řetězec s kvalitou GSM signálu dle momentálního měření. Kvalita signálu je parametr vypočtený ze síly, latence a jiných parametrů, není to jen síla signálu! Tento parametr se pohybuje od 1 do 30, 0 = není signál, 31 = přebuzení signál

Vstupní parametry: žádné

Výstupní parametry: String jméno operátora

Příklad:

**Serial.println(“GSM signal quality:” + getQuality());**

**timeStamp();**

Popis: Vrací řetězec s časovou značkou obdrženou ze sítě GSM, tedy s poměrně přesným časem

Vstupní parametry: žádné

Výstupní parametry: String časová značka

Příklad:

**Serial.println(“GSM time:” + timeStamp());**

Knihovnu lze stáhnout zde. Po její instalaci do prostředí IDE Arduina (nezapomeňte toto prostředí restartovat po nakopírování knihovny) lze přejít k napsání sketche, který se bude chovat dle popisu výše. Využitím této knihovny se dostaneme opravdu k velmi primitivnímu způsobu programování a implementace funkcí Vašeho zařízení bude trvat minuty.

**Ovládání brány prozvoněním**

Uvádím kompletní sketch v celé jeho jednoduchosti, snad netřeba nějakých zvláštních komentářů:

Úvodní definice

#include <ArduinotechGSMShield.h>

#include <SoftwareSerial.h>

#define LED 7

#define RELE1 8

Vytvoření instance třídy AGS

AGS modul(1);

uint8\_t infoStatus;

String number;

Vstupní parametry a inicializace shieldu

void setup()

{

pinMode(RELE1,OUTPUT);

digitalWrite(RELE1,LOW);

pinMode(LED,OUTPUT);

digitalWrite(LED,HIGH);

modul.begin();

}

Hlavní smyčka – čeká na příchod volání, jakmile jej detekuje, hovor zavěsí a načte si číslo volajícího. Je-li shoda s tím, na co čeká, zapne relé, dvě sekundy počká, vypne relé a prozvoní zpět – prozvání 8sekund a pak hovor zavěsí. Není-li shoda, hovor pouze zavěsí bez nějaké další aktivity.

void loop()

{

infoStatus = modul.checkCallAndSMS();

if (infoStatus == 1)

{

number = modul.getNumber();

Serial.println("Call from:" + number);

modul.callEnd();

if (number == "739822476")

{

digitalWrite(RELE1,HIGH);

digitalWrite(LED,LOW);

delay(2000);

digitalWrite(RELE1,LOW);

digitalWrite(LED,HIGH);

modul.makeCall(number);

delay(8000);

modul.callEnd();

}

else modul.callEnd();

}

delay(1000);

}

V dalším dílu seriálu se podíváme na příklady s SMS a hovory tak, abychom maximálně využili potenciálu ArduinoGSMShieldu.