

## Hladinomer - webové rozhranie - Inštrukcie

Ďakujem za zakúpenie projektu Hladinomer, ktorý podporuje využitie otvoreného hardvéru z rady Arduino (AVR), Espressif Systems (ESP8266, ESP32). Projekt sa skladá z webovej aplikácie, ktorá slúži pre zber dát o výške hladiny vody z mikrokontroléru. Dáta dokáže reprezentovať používateľovi, ktorý web aplikáciu používa. Web aplikácia dokáže reprezentovať posledný nameraný údaj v reálnom čase s automatickou aktualizáciou v hlavnom prehľade, ale taktiež aj historicky v tabuľkovom prehľade, kde je možné nájsť posledných 100 záznamov, prípadne historicky všetky dostupné záznamy.

Pre vývoj hladiny vody v čase je možné využiť grafický prehľad s čiarovým grafom, ktorý umožňuje prehľad nameraných údajov za posledné 2 hodiny, 24 hodín, 7 dní, 30 dní a 365 dní. Prostredníctvom budíkových reprezentácií môže používateľ vidieť nameraný rekord minimálny a maximálny za 24 hodín, 7 dní, 30 dní. Mikrokontroler sa stará o zber dát o výške hladiny vody a obsluhuje ultrazvukový senzor vzdialenosti. Pri meraní vyšle ultrazvukový senzor Trigger signál o dĺžke 10 mikrosekúnd, ktorý sa od hladiny odrazí a vráti sa do prijímača - signál Echo. Na základe rýchlosti šírenia zvuku je možné vypočítať vzdialenosť medzi senzorom a hladinou v centimetroch.

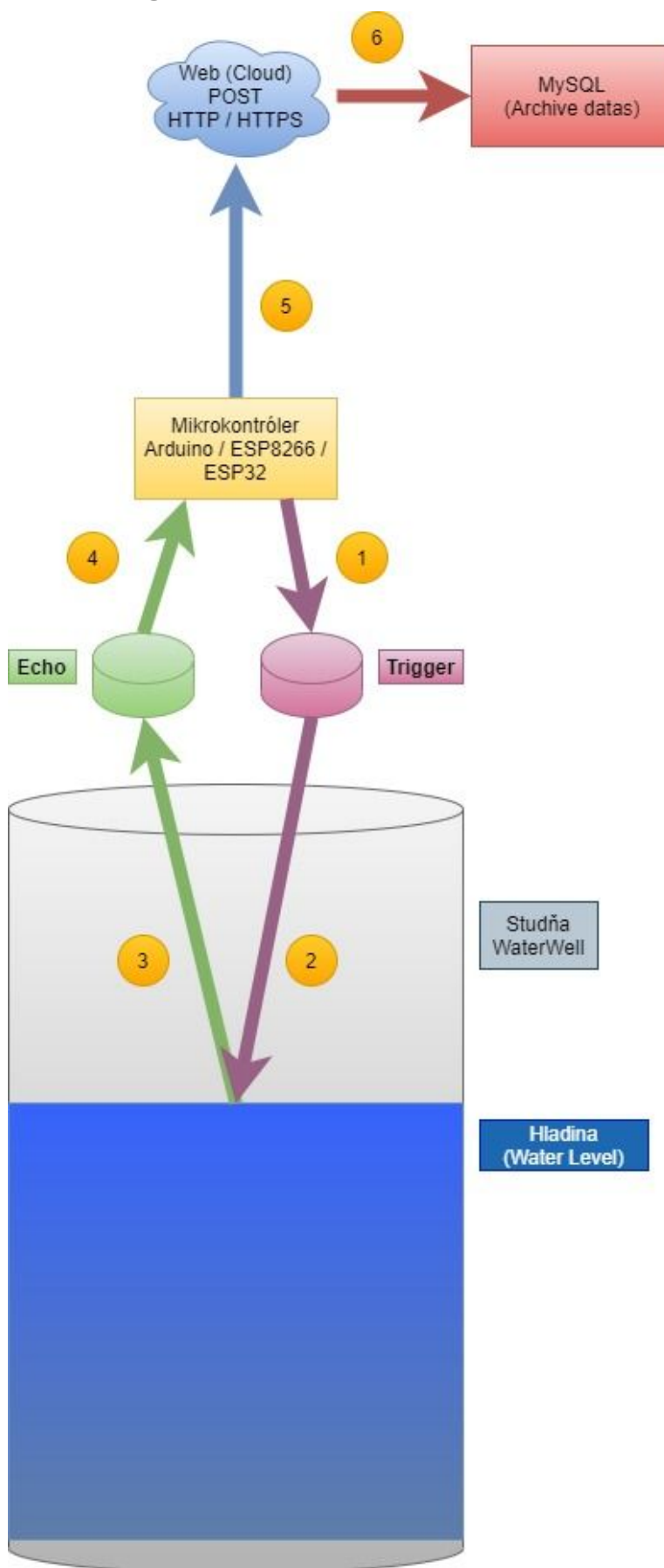
Túto hodnotu odosiela mikrokontroler do webového rozhrania HTTP metódou POST. Dáta môžu byť posielané cez HTTP, ale i HTTPS protokol u platforiem ESP8266 a ESP32, ktoré podporujú šifrovanie. Pre prenos dát je možné použiť prenosovú technológiu Ethernet, WiFi, IoT sieť Sigfox.

**Ultrazvukový senzor vzdialenosti sa umiestňuje kolmo na hladinu a ideálne na stred studne pre optimálne meranie bez odrazov a iných rušivých javov, ktoré by znepresňovali merania, nakoľko má vyžarovaný lúč kruhovú (kuželovú) charakteristiku.** Pre konzistentné výsledky merania sa využíva meranie 10 krát za sebou, ktoré sa priemeruje pre dosiahnutie výslednej - optimálnej hodnoty.

Webové rozhranie vykonáva po prijatí dát z mikrokontroléru korekciu, nakoľko nameraná výška hladiny nereprezentuje skutočnú výšku hladiny od dna, ale iba vzdialenosť od veka studne, kde je senzor umiestnený. Z toho dôvodu je možné do webového rozhrania zadať hĺbku studne, ktorá slúži ako korekcia pre správny prepočet výšky hladiny na skutočnú (od dna) studne. Druhým konfigurovateľným parametrom vo webovom rozhraní je priemer studne, ktorý slúži na možný prepočet výšky hladiny vody v studni na litre. Priemer studne využíva aj trigonometrický - teoretický odhad merateľnej výšky hladiny bez odrazov. Odhad vyjadruje maximálnu hĺbku studne, kedy je možné merať bez odrazov na základe známych detekčných charakteristík (známa šírka lúča) ultrazvukových senzorov vzdialenosti.

Programová implementácia pre mikrokontroléry ráta s využitím ultrazvukových senzorov HC-SR04 (plne kompatibilný i HY-SRF05), alebo JSN-SR04T (vodotesný senzor) v 4-vývodovej verzii (s malou úpravou programu je možné využiť aj 3-vývodovú verziu s jedným vývodom pre Trigger i Echo). Sensory sú schopné merať maximálne do cca 450 cm. **HC-SR04 má detekčnú charakteristiku 15° a je vhodný do väčšiny studní**, nie je však vodotesný a je pri ňom vysoké riziko oxidácie v dôsledku vlhkosti v studni, je ho vhodné umiestniť nad studňu. Vodotesný senzor JSN-SR04T má 60° detekčnú charakteristiku, čo má následok široký lúč, vyžaduje sa tak studňa s dostatočným priemerom (min. 6 metrov v hĺbke 4 a pol metra). **Ultrazvukové senzory nie je možné použiť vo vrtaných studniach, charakteristiky sú príliš veľké!** Oba senzory je nutné napájať na 5V, riadiace signály postačujú aj 3.3V.

## Bloková schéma fungovania hladinomera - princíp:



### Podporovaný hardvér a prenosová technológia:

- **Arduino** Uno / Nano / Mega + **Ethernet** (Wiznet W5100 / W5500 (i v2 USR-ES1), ENC28J60) - podpora HTTP
- **ESP8266** (NodeMCU, Wemos D1 Mini) - vstavaná WiFi - podpora HTTP + HTTPS
- **ESP32** (Devkit, ESP32-WROOM-32) - vstavaná WiFi - podpora HTTP + HTTPS
- Akýkoľvek MCU + **Sigfox** WISOL SFM10R1 UART Modem - podpora HTTP + HTTPS

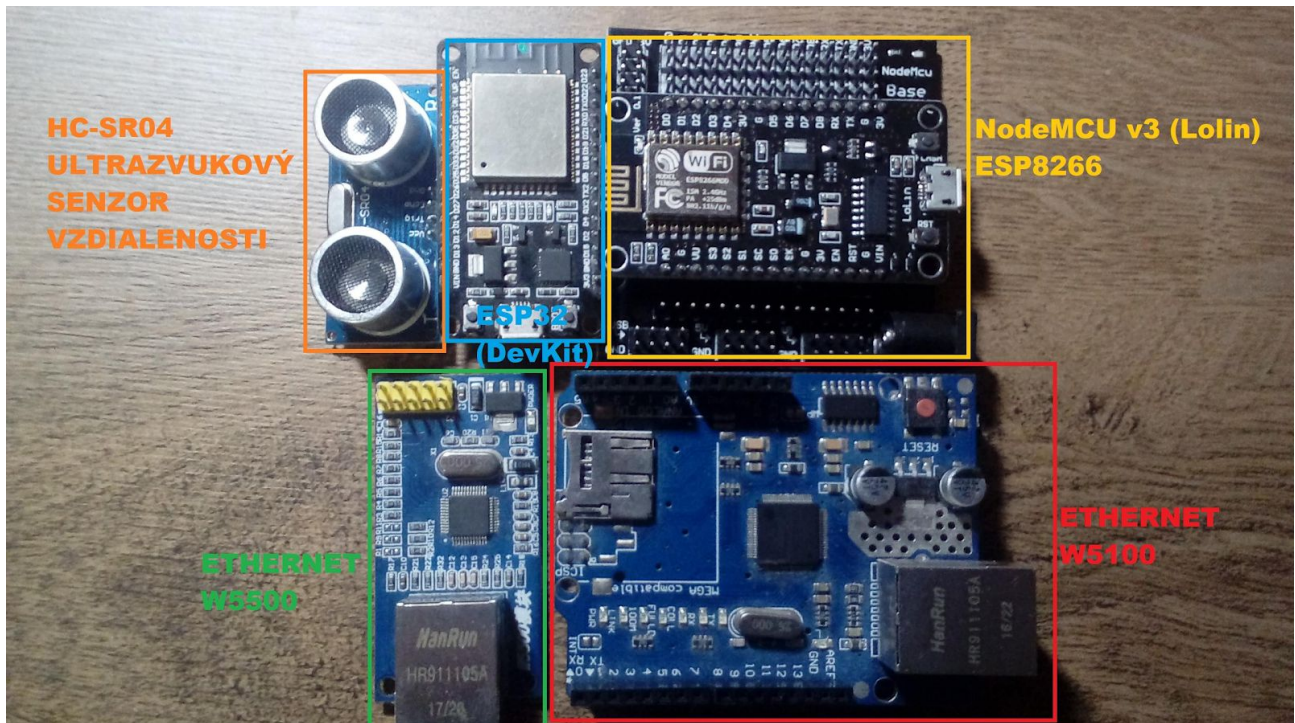


Schéma zapojenia pre Hladinomer je dostupná priamo vo webovom rozhraní. Nakoľko platforma ESP8266 a ESP32 má 3.3V operačnú logiku, je nutné signál Echo pripojiť cez napäťový delič, ktorý zníži logickú úroveň z 5V na 3.3V. **Vývody mikrokontrolérov ESP nie sú 5V tolerantné!** Zapojenie vývodov zo schémy je použité aj v programovej implementácii pre mikrokontroléry, nie je tak nutná žiadna úprava zdrojových kódov. Tie sú dostupné taktiež priamo vo webaplikácii, ktorá vygeneruje ich kód na základe domény, kde hladinomer beží - nájde cestu k .php súboru, ktorý slúži pre zápis údajov do webového rozhrania. Kód tak stačí nakopírovať do prostredia Arduino IDE a nahráť do mikrokontroléru.

V prípade použitia modulu pre Sigfox je potrebné komunikovať s týmto senzorom cez zbernicu UART. Využiť je možné hardvérovú UART linku (napr. UART1, UART2 - ak mikrokontrolér podporuje), alebo emulovať softvérovú cez SoftwareSerial knižnicu - podporuje iba nízke dátové rýchlosti. Správa hladinomeru má veľkosť 4B (INT - celé číslo), zostávajúcich 8B, ktoré je možné v správe použiť je možné využiť pre prenos systémových dát (GPS súradnice, RSSI, číslo správy), nutné nastaviť callback pre automatizovaný zápis údajov zo Sigfox backendu na webové rozhranie.

Pre platformy ESP8266, ESP32 je dostupný prenos údajov aj po HTTPS protokole. Pre realizáciu zabezpečeného spojenia mikrokontrolér ESP32 využíva Root CA certifikát v .pem formáte, ESP8266 SHA1 fingerprint certifikátu webservera. Do programových implementácií je nutné implementovať certifikát, resp. fingerprint manuálne do zdrojových kódov (program.php). Získať fingerprint i Root CA certifikát v .pem formáte je možné cez kryptografický nástroj OpenSSL.

## Prevádzkové režimy mikrokontrolérov:

Programové implementácie pre mikrokontroléry vznikli pre rôzne potreby súvisiace s prevádzkou projektu a možnosti jeho správy. StandBy režim je dostupný pre všetky platformy - t.j. Arduino + Ethernet / ESP8266 / ESP32. V tomto režime je mikrokontrolér v aktívnom operačnom móde, vykonáva requesty, obsluhuje WiFi / Ethernet stack, "kŕmi" watchdog, časuje odosielanie údajov cez millis(). Druhým režimom prevádzky je StandBy + OTA, ktorý je dostupný iba pre platformy ESP8266 a ESP32.

Mikrokontroléry pri nahratí programu OTA sa prevádzkujú ako v režime StandBy, avšak navyše vysielajú sieťový OTA port, ktorým je možné do mikrokontroléru nahráť nový program - firmvér vzdialene - cez LAN sieť priamo z prostredia Arduino IDE. Port sa zobrazí v časti Tools (Nástroje) → COM ports a je v tvare ESPXX-MAC adresa at IP\_ADRESA.

Aby bola OTA funkcionálna dostupná aj po nahratí nového programu, musí v sebe obsahovať Basic OTA implementáciu, v opačnom prípade sa prestane vysielat' OTA sieťový port a nebude ju možné vzdialene aktualizovať. Pri sieťovom OTA porte nie je možné zobrazit' UART (Sériový) monitor. **Pri zariadeniach typu Windows je nutné vypnúť bránu Firewall**, nakoľko blokuje odpoveď mikrokontroléru smerom do počítača a nahratie programu sa nepodarí s chybou "ESP timed out".

Zariadenie, ktoré realizuje OTA aktualizáciu môže byť do LAN siete pripojené cez Ethernet / WiFi. Okrem prostredia Arduino IDE je nutné mať nainštalovaný aj nástroj Python minimálne vo verzii 3. Python automatizovane riadi aktualizáciu a prenos údajov. Platforma ESP musí mať tabuľku partícií s možnosťou OTA pre kompatibilitu so spôsobom aktualizácie.

Pre spomenuté platformy z rady Espressif Systems je možné použiť ešte jeden prevádzkový mód a to ULP - Ultra Low Power. Tento prevádzkový mód používa hlboký spánok hlavného procesora, čím dramaticky zníži spotrebu elektrickej energie, rádovo na 10 mikroampérov pri 3.3V napájaní. Tento režim je tak vhodný pre prevádzku hladinomera na batériu. Do programovej implementácie je možné doplniť aj možnosť ovládania tranzistora, ktorým je možné spínať napájanie ultrazvukového senzora, nakoľko ten má prúdový odber 1mA pri 5V napájaní v StandBy stave.

Pri uspatí hlavného procesora sa nastaví spôsob jeho prebudenia po 300 sekundách (5-tich minútach), kedy chceme posielat' dáta na webserver a vykonať meranie zo senzora. Platforma ESP8266 využíva zdroj prebudenia Timer. Časovač vyšle z GPIO16 (D0) vývodu WAKE signál (GND) pripojený na RST. Tento signál reštartuje mikrokontrolér a spustí hlavný čip - toto zapojenie je obsiahnuté v schéme zapojenia vo web aplikácii. Ak je prepojený D0 s RST, nie je možné nahráť nový program! Pri nahrávaní programu je nutné túto prepojku rozpojiť a po nahratí programu zapojiť späť, inak sa mikrokontrolér nezobudí z režimu spánku.

Platforma ESP32 umožňuje taktiež prebudenie prostredníctvom časovača, avšak využíva RTC Timer, teda časovač z hodín reálneho času. Tento časovač je presnejší v porovnaní s ESP8266 a nevyžaduje prepojku, nakoľko časovač vykoná reštart ESP32 systémovým volaním. U ESP8266 je možný rozptyl časovača na úrovni 10 až 15%.

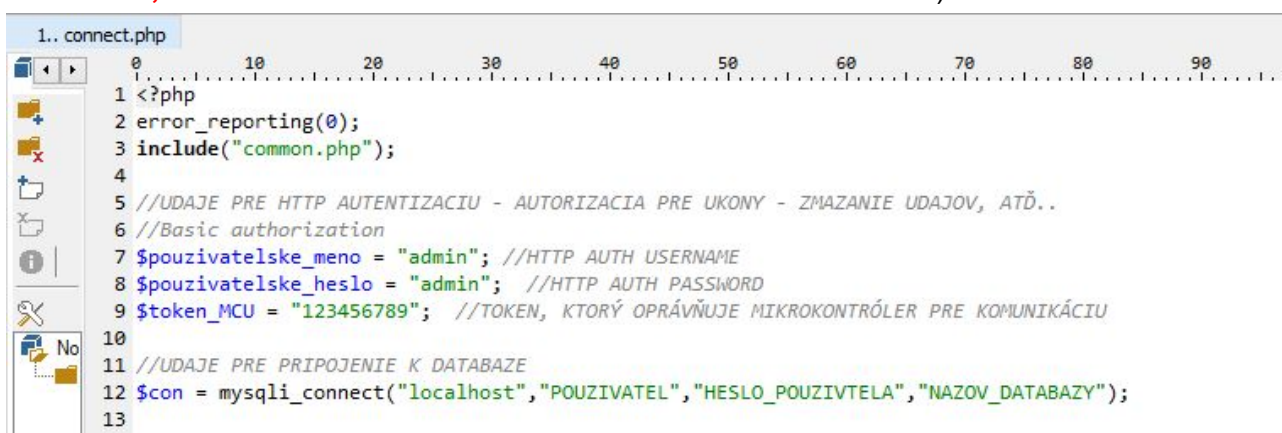


## Príklady možného umiestnenia senzorov do studne



## Inštalácia a uvedenie systému Hladinomer do prevádzky:

1. Rozbaľte .zip archív projektu Hladinomer
2. Prihláste sa do vášho rozhrania pre správu MySQL databáz (PHPMyAdmin)
3. Zvoľte si databázu do ktorej chcete importovať tabuľku projektu so štruktúrou
4. Kliknite v hornej lište na Importovať a zvoľte súbor obsiahnutý v priečinku /sql/
5. Otvorte súbor connect.php a zmeňte údaje k vašej MySQL databáze (host, používateľské meno, heslo a názov databázy, kde ste importovali tabuľku studna2 z predmetného .sql súboru)
6. Taktiež upravte token, ktorý bude oprávňovať váš mikrokontrolér posilať dáta na server, môže byť tvorený znakmi, číslicami (napríklad: dsf4ds98f4s9df9). Zmeňte používateľské meno a heslo pre HTTP Basic Autentizáciu, ktorá vás bude oprávňovať vykonávať zmeny s dátami (**POZOR, TIETO PRIHLASOVACIE INFORMÁCIE NIE SÚ ŠIFROVANÉ! ODOSIELA SA ICH BASE64 HODNOTA! HODNOTA SA PRENÁŠA ŠIFROVANÝM KANÁLOM, IBA AK KLIENT PRISTUPUJE K SERVERU CEZ HTTPS!**)



```
1. connect.php
0      10      20      30      40      50      60      70      80      90
1 <?php
2 error_reporting(0);
3 include("common.php");
4
5 //UDAJE PRE HTTP AUTENTIZACIU - AUTORIZACIA PRE UKONY - ZMAZANIE UDAJOV, ATD..
6 //Basic authorization
7 $pouzivatelske_meno = "admin"; //HTTP AUTH USERNAME
8 $pouzivatelske_heslo = "admin"; //HTTP AUTH PASSWORD
9 $token_MCU = "123456789"; //TOKEN, KTORÝ OPRÁVŇUJE MIKROKONTROLÉR PRE KOMUNIKÁCIU
10
11 //UDAJE PRE PRIPOJENIE K DATABÁZE
12 $con = mysqli_connect("localhost","POUZIVATEL","HESLO_POUZIVATELA","NAZOV_DATABAZY");
13
```

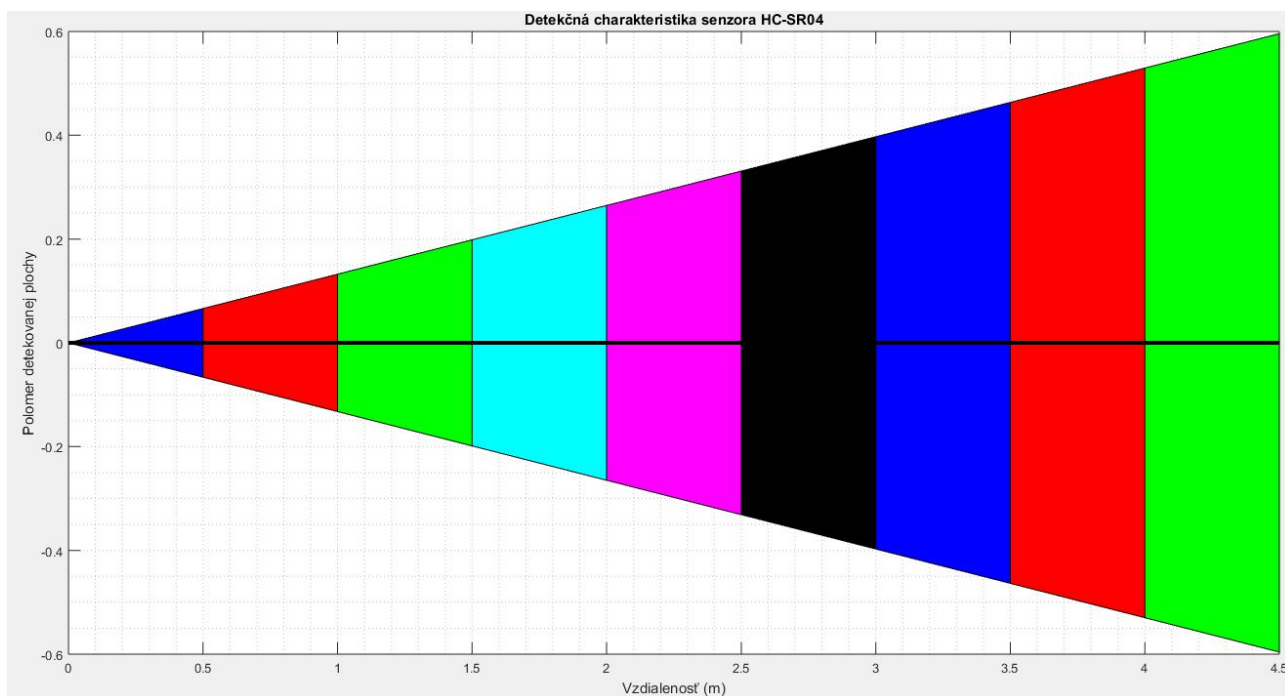
7. Skopírujte všetky súbory a priečinky (sql priečinkov netreba) na váš webserver do priečinka www (môže sa kopírovať do samostatnej zložky pod hlavnou www ako samostatný projekt, napríklad /hladinomer/súbory...)
8. Webové rozhranie je pripravené na prevádzku a záznam dát
9. Stiahnite si Github repozitár: <https://github.com/martinius96/hladinomer-studna-scripty>, kliknite v pravej časti na zelené tlačidlo Code → Download ZIP
10. Archív rozbaľte a obsah zložky /src/ (Celé priečinky Ethernet2, Ethernet3, NewPing, NewPingESP8266, UIPEthernet) nakopírujte do:  
**C:/Users/[Aktuálny\_používateľ]/Dokumenty/Arduino/libraries** - Windows 10 - preddefinované umiestnenie (môže sa líšiť na základe toho, kde máte nainštalované Arduino IDE v systéme!) - jednotlivé priečinky sú knižnice, ktoré sú potrebné pre podporovaný hardvér v systéme a pre úspešnú kompiláciu programu.
11. Otvorte webové rozhranie, prejdite na podstránku Program, prihláste sa do systému cez HTTP Auth (meno a heslo ste zadali do connect.php), po pridelení prístupu môžete vidieť zdrojový kód pre váš mikrokontrolér.
12. Program skopírujte do Arduino IDE, zvoľte cieľovú dosku, COM port, kde je doska pripojená, ak je nutné zadajte v prípade WiFi platformy meno domácej WiFi siete a heslo, odkomentujte knižnicu a zakomentujte pôvodnú (rozdielne knižnice pre rozdielne Ethernet shiedly a moduly), kliknite na Upload (Nahráť program) - **časti programovej implementácie sú vyznačené červenou farbou, ak potrebujú vašu pozornosť / zmenu na základe pripojeného hardvéru pre jednoduchú orientáciu**
13. Po nahratí programu začne mikrokontrolér odosielať dáta v 5 minútových intervaloch, respektíve v prípade využitia Sigfox modemu je to 11 minút

## Problémy, podpora hardvéru:

- Vývojové platformy ESP8266 a ESP32 nie sú prostredím Arduino IDE štandardne podporované, je nutné doinštalovať ich podporu cez Manažér dosiek v prostredí Arduino IDE podľa návodu:
  - **ESP8266** - <https://github.com/esp8266/Arduino#installing-with-boards-manager> - zvoliť 2.7.4. (poslednú stable verziu, pre ktorú sú kompatibilné programové implementácie Hladinomera)
  - **ESP32** - [https://github.com/espressif/arduino-esp32/blob/master/docs/arduino-ide/boards\\_manager.md](https://github.com/espressif/arduino-esp32/blob/master/docs/arduino-ide/boards_manager.md) - kompatibilná akákoľvek verzia od 1.0+
- Ak sa v Arduino IDE po pripojení dosky neukazuje COM port (voľba je šedá a neumožňuje výber), respektíve je nastavený stále na COM1, je nutné nainštalovať ovládač daného USB-UART prevodníka, **prípadne skontrolujte, že používate USB kábel s dátovými vodičmi, niektoré káble majú iba napájanie bez dátových vodičov!**
- Vývojová platforma ESP8266 (NodeMCU, Wemos D1 Mini) je najčastejšie osadená USB-UART prevodníkom CH340, ktorý je nutné manuálne doinštalovať: <https://sparks.gogo.co.nz/ch340.html>
- ESP32 (Devkit) obsahuje najčastejšie USB-UART prevodník CP2102 od Silicon Labs. Prevodník je nutné manuálne nainštalovať z oficiálnych stránok Silicon Labs: <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers> - tento typ prevodníka sa vyskytuje aj u niektorých ESP8266 dosiek, napríklad u NodeMCU v2 (Amica) kvôli kompaktným rozmerom
- **MacOS používatelia** - Ak hlási kompilátor chybu s funkciami PROGMEM, je nutné ich manuálne odstrániť z programu, reťazce však budú priradené do RAM pamäte mikrokontroléru
- ESP32 Devkity majú problémovú konštrukciu s dvoma identickými kondenzátormi pre signál EN a BOOT (GPIO 0) a doska nenabootuje. Pri štarte systému (power-on cyklus, pripojenie napájania) sa z dôvodu rovnakých - GND potenciálov na oboch vývodoch prepne mikrokontrolér do download módu a čaká na manuálny reštart tlačidlom EN (RESET). Oprava je možná pripojením GPIO 0 cez 10K pullup rezistor na Vcc (3.3V), prípadne je možné odpájať kondenzátor, alebo nahradiť iným s inou kapacitou - známa chyba Devkitov.
- Ak webserver nezobrazuje stránku, ale iba texty programového kódu, je nutné doinštalovať Apache / Nginx, prípadne HTTPD službu, ktorá umožňuje spúšťať backend - server side scripty v PHP - Je možné využiť PHP vo verzii 5.6+ až po najnovšie verzie PHP
- Ak na webstránke projektu vidíte informáciu - "Failed to connect to DataBase", nezadali ste správne informácie o vašej databáze v súbore connect.php
- Pre získanie Root CA certifikátu nástrojom OpenSSL použijeme príkaz `openssl s_client -showcerts -verify 5 -connect myserver.com:443 < /dev/null` - toto riešenie vydrží v implementácii rádovo 5 až 20 rokov v závislosti od dĺžky platnosti certifikátu certifikačnej authority (samozrejme ak je certifikát pre webserver vydaný po čase inou autoritou, treba ju zmeniť, pripojenie sa inak nevykoná)
- Pre získanie fingerprintu certifikátu servera v SHA1 formáte nástrojom OpenSSL použijeme príkaz `openssl s_client -connect myserver.com:443 -showcerts < /dev/null 2>/dev/null | openssl x509 -in /dev/stdin -sha1 -noout -fingerprint` - toto riešenie vydrží v implementácii rádovo 1 až 2 roky, čo je doba platného certifikátu webservera, vždy keď sa vykoná jeho renew sa zmení aj fingerprint, ktorý je potrebné opäť implementovať do zdrojového kódu pre mikrokontrolér ESP8266



## Detekčná charakteristika HC-SR04:



## Detekčná charakteristika JSN-SR04T:

