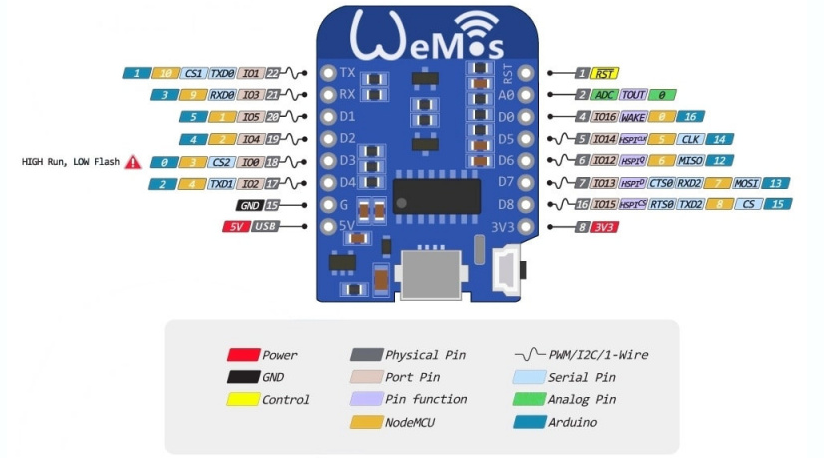
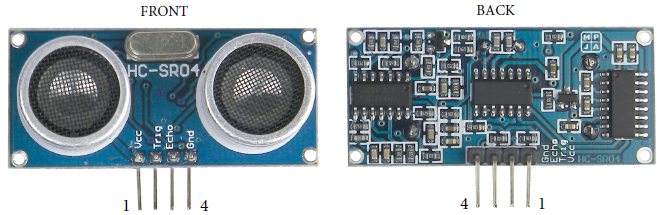
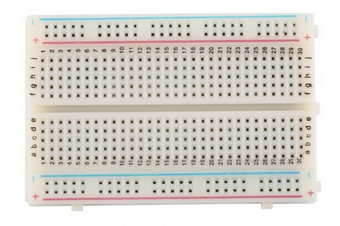
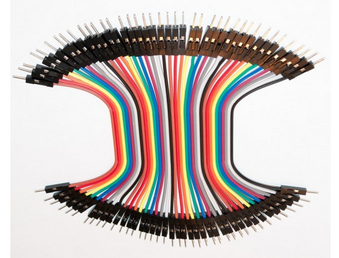
# Komponenty

* **WeMos D1 Mini ESP8266 WiFi modul**https://www.laskakit.cz/wemos-d1-mini-esp8266-wifi-modul/  
    
  
* **Ultrazvukový měřič vzdálenosti HC-SR04**https://www.laskakit.cz/arduino-ultrazvukovy-meric-vzdalenosti-hc-sr04/


* **Nepájivé kontaktní pole 400 pinů, Bílé**https://www.laskakit.cz/nepajive-kontaktni-pole-400-pinu--bile/  
  
* **Dupont propojovací kabel 40žil M-M samec-samec (10cm)**https://www.laskakit.cz/dupont-propojovaci-kabely-40ks-m-m-samec-samec--10cm-/  
  
* **Síťový napájecí adaptér 5V/3A pro Raspberry Pi microUSB**https://www.laskakit.cz/napajeci-adapter-sitovy-3a-5v-pro-raspberry-pi-microusb/  
  
* **Univerzální instalační krabice ABS IP65 100x100x75mm**https://www.laskakit.cz/univerzalni-instalacni-krabice-abs-ip65-100x100x75mm/  
  

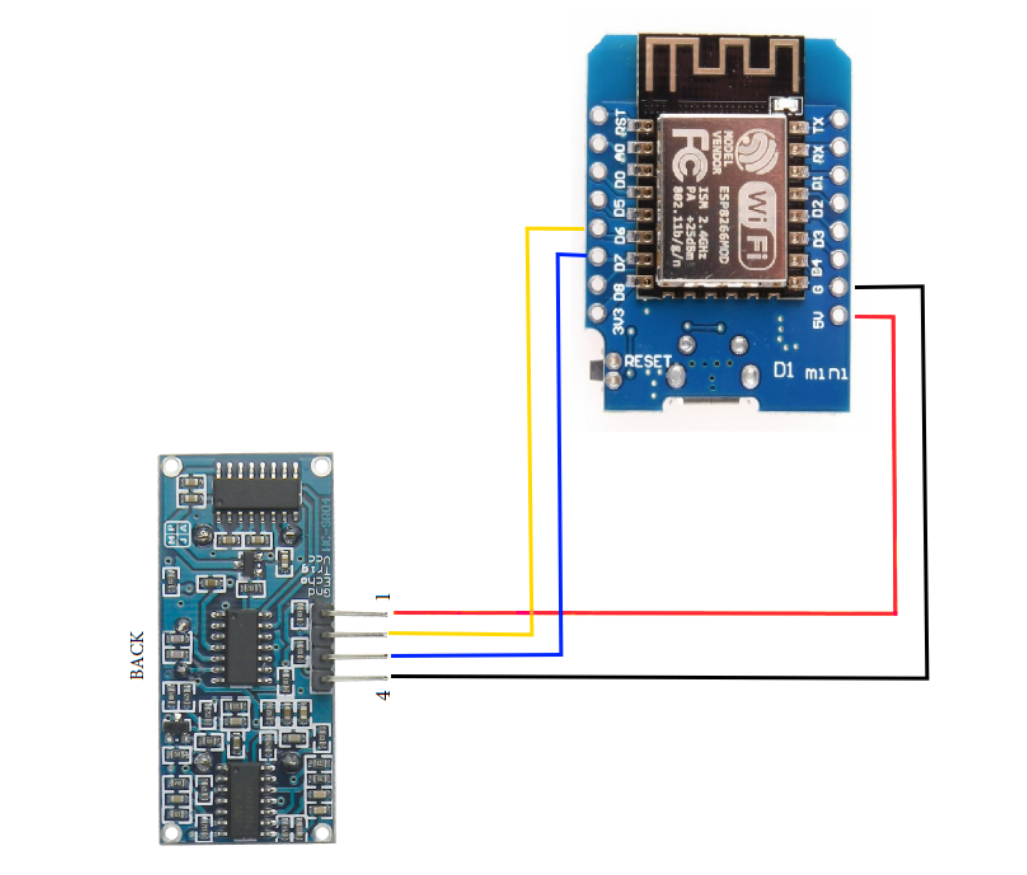
# Oživenie

* postupovali sme podľa návodu z eShopu priloženého k tomuto modulu  
  <https://blog.laskakit.cz/zaciname-s-wemos-d1-mini-s-wifi-modulem-esp8266/>  
  ktorý obnášal:
  + stiahnutie a inštaláciu SW prostredia pre vývoj Arduino IDE na PC
  + stiahnutie a inštalácia ovládača USB prevodníka pre chip CH340
  + spojazdnenie podpory čipu ESP8266 v Adruino IDE  
    (port sa objaví v Arduino IDE po pripojení modulu cez USB k PC)
  + nahratie testovacieho programu do modulu

# Zapojenie

Zapojenie sme vykonali podľa nižšie uvedenej schémy – zem (GND) sme prepojili čiernym vodičom a výstup 5V, ktorý poskytuje modul WeMos Mini, sme využili na napájanie detektora (VCC) a prepojili ho červeným vodičom. Vstup (Trig) a výstup (Echo) detektora sme pripojili na digitálne I/O porty modulu D6 a D7 žltým a modrým vodičom.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **WeMos Piny** | |  | **HC-SR04 piny** | | |
| **PIN** | **ESP8266 mapovanie** | **ID** | **PIN** | **POPIS** |
| 5V | n/a |  | 1 | VCC | 5V power supply |
| D6 | I/O, MISO, GPIO12 |  | 2 | Trig | Trigger Input pin |
| D7 | I/O, MOSI, GPIO13 |  | 3 | Echo | Receiver Output pin |
| G | Ground |  | 4 | GND | Power ground |



# Prvý kód

Cyklicky vypisuje v “Serial monitor” okne prostredia Arduino IDE vzdialenosť objektu od detektora v cm.

Senzor HC-SR04 umožňuje odmerať čas, za ktorý sa vráti odrazený ultrazvukový signál, ktorý predtým vyslal. Tento čas sa vráti mikrosekundách (µs), ktoré keď vynásobíme vhodnou konštantou získame vzdialenosť prekážky. Z rýchlosti zvuku sme vypočítali koľko mu trvá dráha 1cm – vyšlo nám 56.2 µs – a toto je konštanta, ktorou keď podelíme získaný čas, dostaneme vzdialenosť prekážky v cm.

Nižšie nasleduje postup, ktorým sme určili túto konštantu:

* rýchlosť zvuku je 343.2 m/s
* signál sa musí ale od prekážky odraziť a vrátiť späť, takže prekonáva dvojnásobnú vzdialenosť, čo je možné vyjadriť aj ako keby mal polovičnú rýchlosť a teda   
  343.2/2 = 172.6 m/s
* po prevode jednotiek na z metrov na cm  
  172.6 m/s = 17160 cm/s
* po prevode jednotiek zo sekúnd na µs  
  17160 cm/s = 0.01716 cm/µs
* po otočení rýchlosti dostaneme čas, ktorý trvá prejdenie 1cm  
  1cm / 0.01716 cm/µs = 58.2 µs



#define trigPin D6 // vysielac

#define echoPin D7 // prijimac

long duration, distance; // premenne pouzite na urcenie vzdialenosti

const float onecmtime = 58.2; // konstanta casu potrebneho na 1cm

void setup()

{

  Serial.begin (9600);

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

}

void loop()

{

  /\* reset vysielaca - ticho \*/

  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  /\* vysielac vysiela 10ms \*/

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  /\* vypnutie vysielaca\*/

  digitalWrite(trigPin, LOW);

  /\* cakame na odozvu \*/

  duration = pulseIn(echoPin, HIGH);

  /\* urcenie vzdialenosti v cm podla rychlosti zvuku za ktoru prejde 1cm \*/

  distance = duration/onecmtime;

  /\* vystup do seriovy monitor \*/

  Serial.println(distance);

  /\* pauza 50ms pre dalsim koleckom \*/

  delay(50);

}

# Druhý kód

Vyskúšali sme sa pripojiť s modulom na WiFi sieť a spustiť na ňom lokálny webserver, pomocou ktorého vieme zobraziť jednoduchú webstránku. Modul dostane na sieti pridelnú IP adresu a po jej zadaní do browsera na inom zariadení pripojenom k tej istej sieti, sa nám zobrazí pripravená webstránka.



#include <ESP8266WiFi.h>//zahrnie kniznicu na pracu s WiFi modulom

#include <ESP8266WebServer.h> //zahrnie kniznicu na pracu s webserverom

const char\* nazov ="Gries"; //konstanta s nazvom WiFi siete

const char\* heslo ="12345678"; //konstanta s heslom do WiFi siete

ESP8266WebServer server(80); //cislo 80 je predvoleny port pre HTTP

void setup()

{

  Serial.begin (9600);

  Serial.print("Pripajam sa k sieti... ");

  Serial.println(nazov); //vypise nazov siete, ktory sme uviedli vyssie

  WiFi.begin(nazov,heslo); //pripaja sa do lokalnej WiFi siete

  while (WiFi.status()!=WL\_CONNECTED)

    delay(1000); //pockame kym bude wifi pripojena

  server.on("/", pripojeny); //obsluzna funkcia ak webserver bezi

  server.onNotFound(nepripojeny); //obsluzna funkcia ak nebezi

  server.begin(); //spusti server

}

void loop()

{

  server.handleClient();

}

void nepripojeny()

{

  server.send(404, "text/plain", "Server nebezi");

}

//funkcia, ktora spracovava dotazy na webserver a odpoveda HTML strankou

void pripojeny()

{

  String ptr = "";

  ptr +="<!DOCTYPE html>\n";

  ptr +="<html>\n";

  ptr +="  <head>\n";

  ptr +="    <title>nazov stranky</title>\n";

  ptr +="  </head>\n";

  ptr +="  <body>\n";

  ptr +="    <p><strong>Vzdialenost hladiny zumpy je XXX cm.</strong></p>\n";

  ptr +="  </body>\n";

  ptr +="</html>\n";

  server.send(200, "text/html", ptr);

}

# Treti kód

Nakoniec sme skombinovali predošlé 2 kódy aby sme pri požiadavke na zobrazenie web stránky odmerali vzdialenosť senzorom a vygenerovali stránku s výsledkom, ktorá sa zobrazí volajúcemu.



#include <ESP8266WiFi.h>//zahrnie kniznicu na pracu s WiFi modulom

#include <ESP8266WebServer.h> //zahrnie kniznicu na pracu s webserverom

#define trigPin D6 // vysielac

#define echoPin D7 // prijimac

long duration, distance; // premenne pouzite na urcenie vzdialenosti

const float onecmtime = 58.2; // konstanta casu potrebneho na 1cm

const char\* nazov ="Gries"; //konstanta s nazvom WiFi siete

const char\* heslo ="12345678"; //konstanta s heslom do WiFi siete

ESP8266WebServer server(80); //cislo 80 je predvoleny port pre HTTP

void setup()

{

  Serial.begin (9600);

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

  Serial.print("Pripajam sa k sieti... ");

  Serial.println(nazov); //vypise nazov siete, ktory sme uviedli vyssie

  WiFi.begin(nazov,heslo); //pripaja sa do lokalnej WiFi siete

  while (WiFi.status()!=WL\_CONNECTED)

    delay(1000); //pockame kym bude wifi pripojena

  server.on("/", pripojeny); //obsluzna funkcia ak webserver bezi

  server.onNotFound(nepripojeny); //obsluzna funkcia ak nebezi

  server.begin(); //spusti server

}

void loop()

{

  server.handleClient();

}

void nepripojeny()

{

  server.send(404, "text/plain", "Server nebezi");

}

//funkcia, ktora spracovava dotazy na webserver a odpoveda HTML strankou

void pripojeny()

{

  /\* reset vysielaca - ticho \*/

  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);

  /\* vysielac vysiela 10ms \*/

  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);

  /\* vypnutie vysielaca\*/

  digitalWrite(trigPin, LOW);

  /\* cakame na odozvu \*/

  duration = pulseIn(echoPin, HIGH);

  /\* urcenie vzdialenosti v cm podla rychlosti zvuku za ktoru prejde 1cm \*/

  distance = duration/onecmtime;

  /\* vystup na consolu \*/

  Serial.println(distance);

  String ptr = "";

  ptr +="<!DOCTYPE html>\n";

  ptr +="<html>\n";

  ptr +="  <head>\n";

  ptr +="    <meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no\">\n";

  ptr +="    <meta name=\"HandheldFriendly\" content=\"true\">\n";

  ptr +="    <title>Vzdialenost hladiny</title>\n";

  ptr +="  </head>\n";

  ptr +="  <body style=\"background-image: url('https://rb.gy/tx0zbd'); background-repeat: no-repeat; background-size: 100% auto;\">\n";

  ptr +="    <p style=\"text-align: center; font-size: 100px\">" + String(distance) + " cm</p>\n";

  ptr +="  </body>\n";

  ptr +="</html>\n";

  server.send(200, "text/html", ptr);

}

# Výsledok

Na mobilnom zariadení je možné priebežne kontrolovať stav naplnenia septiku. Užívateľovi sme uložili na plochu mobilného zariadenia odkaz na stránku s pevnou IP adresou, ktorú môže otvárať v lokálnej sieti. Aby sme zabezpečili, že DHCP nepridelí inú adresu, nastavili sme na wifi routeri pevnú adresu pre tento esp modul (podľa jeho MAC adresy).

