

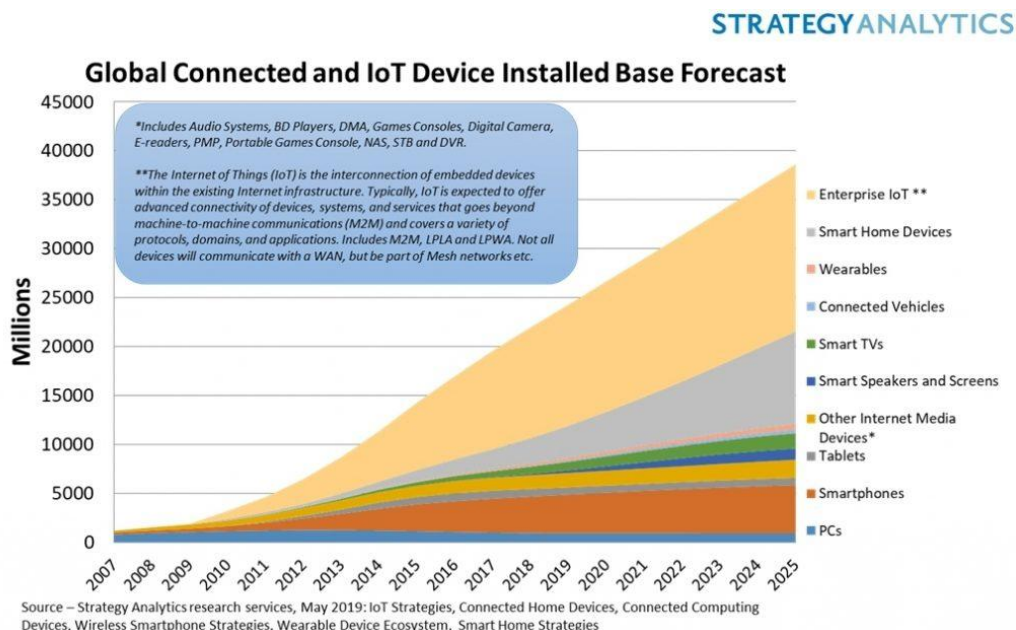
## 9. Urządzenia Internetu Rzeczy

Koncepcja Internetu rzeczy zakłada tworzenie sieci urządzeń (rzeczy), które mogą komunikować się przez internet. Rzeczami internetowymi mogą być obiekty fizyczne wyposażone w interfejs komunikacyjny i jeden lub kilka następujących elementów

- czujniki (temperatury, zanieczyszczeń, ciśnienia, wilgotności, położenia, drgań itp.),
- elementy wykonawcze (przełączniki, sterowane zawory, silniki itp.),
- elementy sygnalizacyjne,
- elementy obliczeniowe.

Internet rzeczy jest systemem obiektów fizycznych podłączenia do globalnego internetu, które można odkrywać, monitorować i kontrolować za pośrednictwem różnych interfejsów sieciowych. Czujniki i moduły komunikacji sieciowej stają się coraz tańsze i mogą być wbudowywane w coraz większą ilość obiektów fizycznych.

Według raportu Davea Evansa (The Internet of Things How the Next Evolution of the Internet is Changing Everything, Cisco IBSG 2011) [7] w 2008 roku liczba rzeczy podłączonych do Internetu przewyższyła liczbę mieszkańców Ziemi. Prognoza Evansa na rok 2020 to 50 miliardów urządzeń podłączonych do internetu. Nowsze analizy między innymi Strategii Analitycznych szacują, że w roku 2020 do internetu podłączonych jest 25 miliardów urządzeń. Wszystkie prognozy są zgodne, że liczba urządzeń podłączonych do Internetu stale rośnie i będzie rosła.



Rys. 9.1. Prognoza ilości urządzeń podłączonych do internetu

Projekt „SezAM wiedzy, kompetencji i umiejętności” jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

Według Strategii Analitycs liczba smartfonów z dostępem do internetu wielokrotnie przewyższa dziś liczbę komputerów PC. Największą grupą urządzeń podłączonych do internetu są urządzenia Internetu rzeczy wykorzystywane w biznesie (Enterprise IoT), drugą w kolejności grupę stanowią inteligentne urządzenia domowe (Smart Home Devices).

Aby rzeczy internetowe mogły się łączyć z internetem muszą być wyposażone w moduły zapewniające interfejs komunikacyjny. Bardzo popularne są moduły Wi-Fi niewymagające połączenia przewodowego.

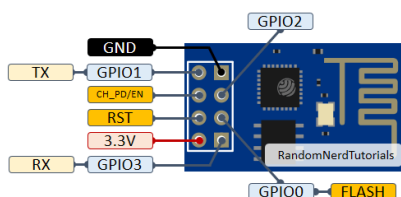
## Popularne moduły Wi-Fi

Istotny wpływ na rozwój Internetu rzeczy miało pojawienie się 30 grudnia 2013r taniego chińskiego układu scalonego ESP8266 wyprodukowanego przez firmę Espressif Systems [5].

### Podstawowe parametry ESP8266

- Napięcie zasilania 3,3 V
- Wbudowany 32 bitowy mikroprocesor RISC Tensilica L106 o częstotliwości taktowania 80 lub 160 MHz
- Pamięć:
  - 32 KiB instruction RAM
  - 32 KiB instruction cache RAM
  - 80 KiB user-data RAM
  - 16 KiB system-data RAM
- Zewnętrzna pamięć Flash do 16 MiB (typowo od 512 KiB do 4 MiB)
- Wi-Fi IEEE 802.11 b/g/n
- 16 pinów ogólnego przeznaczenia GPIO
- Interfejsy: SPI, I<sup>2</sup>C (realizacja softwarowa), I<sup>2</sup>S i UART
- 10 bitowy przetwornik A/C

W sierpniu 2014 na rynku pojawił się wyprodukowany przez Ai-Thinker moduł Wi-Fi ESP-01 wykorzystujący układ ESP8266. Obecnie moduł ten można nabyć w Polsce za około 10zł. Moduł posiada wbudowaną antenę i 8 wyprowadzeń, rysunek 9.2.



Rys. 9.2. Schemat wyprowadzeń modułu ESP-01[16]

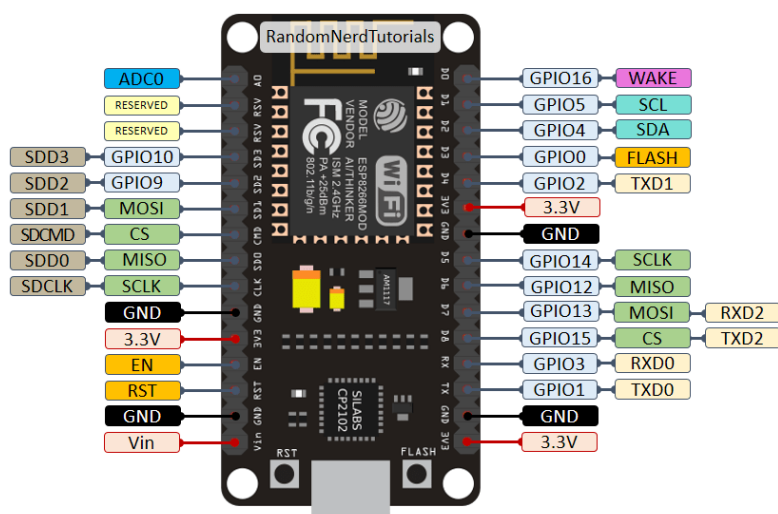
Moduł ESP-01 (niebieski) ma wbudowaną pamięć Flash o pojemności 512 KiB, kolejny moduł ESP-01 (czarny) i ESP-01S mają wbudowaną pamięć Flash o pojemności 1 MiB. Firma Ai-Thinker w kolejnych latach wypuściła na rynek kilkanaście modułów z ESP8266 różniących się ilością pamięci Flash i ilością wyprowadzonych pinów GPIO.

Projekt „SezAM wiedzy, kompetencji i umiejętności” jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

ESP-01 dał początek wielu projektom Internetu rzeczy, świetnie się nadaje do montowania w urządzeniach, ale jest on mało przyjazny dla do nauki i rozwijania oprogramowania. Oprogramowanie modułu jest wgrywane do pamięci Flash za pomocą konwertera USB-UART, co jest dość kłopotliwe w warunkach laboratoryjnych. Wygodniejsze w eksperymentowaniu i nauce są płytki rozwojowe zawierające już w sobie konwerter USB-UART. Jedną z najbardziej popularnych jest moduł ESP8266 NodeMCU WiFi DevKit.

### NodeMCU WiFi DevKit

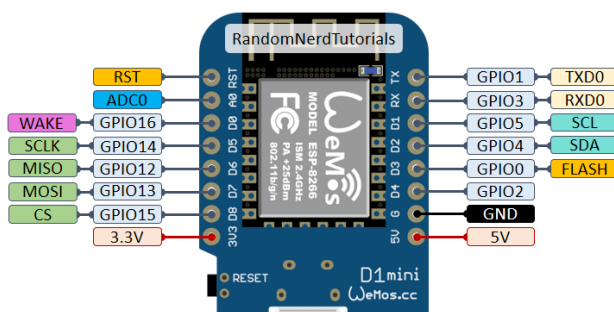
NodeMCU WiFi DevKit jest modulem przeznaczonym do rozwijania oprogramowania dla ESP8266. Moduł ma wbudowany konwerter USB-UART umożliwiający programowanie układu przez złącze micro USB. Złącze micro USB służy także do zasilania układu. NodeMCU WiFi DevKit wykorzystuje moduł ESP-12 w ekranowanej obudowie. Schemat wyprowadzeń NodeMCU WiFi DevKit przedstawiony jest na rysunku 9.3.



Rys. 9.3. Schemat wyprowadzeń NodeMCU WiFi DevKit [16]

NodeMCU występujące w nazwie modułu to nazwa wgranego do pamięci Flash oprogramowania układowego. Oprogramowanie układowe pod nazwą NodeMCU po raz pierwszy zostało opublikowane na GitHubie w październiku 2014r. NodeMCU zostało stworzone w języku Lua i umożliwia programowanie ESP8266 w tym języku. Oprogramowanie było początkowo przeznaczone dla NodeMCU WiFi DevKit, obecnie może być uruchamiane na innych modułach. Najnowsza wersja jest dostępna na GitHubie pod adresem <https://github.com/nodemcu/nodemcu-firmware>.

## WeMos D1 Mini Pinout



Rys. 9.4. Schemat wyprowadzeń modułu WeMos D1 Mini [16]

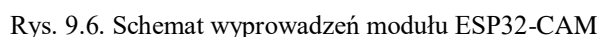
## ESP 32

W sierpniu 2016r. na rynku pojawił się mikrokontroler ESP32, następca ESP8266.

### Podstawowe parametry ESP32

- 32-bitowy mikroprocesor LX6, o częstotliwości 160 lub 240 MHz
- Pamięć 520 KiB SRAM
- Zewnętrzna pamięć Flash do 32 MiB (typowo 4 MiB)
- Wi-Fi IEEE 802.11 b/g/n
- Bluetooth v4.2
- Interfejsy:  $4 \times \text{SPI}$ ,  $2 \times \text{I}^2\text{C}$ ,  $2 \times \text{I}^2\text{S}$  i  $3 \times \text{UART}$
- 12-bitowy przetwornik A/C – 18 kanałowy
- 2 8-bitowe przetworniki C/A
- 10 czujników dotykowych (piny GPIO)
- Czujnik Halla

Wraz z pojawieniem się nowego mikrokontrolera powstały moduły Wi-Fi z tym mikrokontrolerem. Moduły mają wyprowadzenia do montażu powierzchniowego i trudno jest je stosować w celach laboratoryjnych. Do celów badawczo rozwojowych powstały liczne płytki rozwojowe, podobne do NodeMCU WiFi Devkit, zawierające wlutowany moduł ESP32, wyprowadzone piny i konwerter USB-UART. Do najpopularniejszych należą ESP32-DevKit produkcji Espressif i ESP32-CAM produkcji Ai-Thinker. Płytką rozwojową ESP32-CAM zawiera złącze do podłączenia kamery – zwykle jest sprzedawana z kamerą OV2640. Kamera jest niskiej jakości, ale całość można kupić za 7\$, w Polsce od 40zł. Schemat wyprowadzeń ESP32-DevKit przedstawiono na rysunku 9.5 a schemat wyprowadzeń ESP32-CAM na rysunku 9.6.



## 75



programów. Przykładowe programy na mikrokontrolery ESP publikowane na GitHubie są głównie kodowane w Arduino IDE.

Arduino IDE wykorzystuje język programowania Arduino przypominający język C. Program po skompilowaniu jest wgrywany do urządzenia przez konwerter USB-UART. Na części płytek rozwojowych mikrokontrolerów ESP konwerter USB-UART jest wbudowany w płytkę i wystarczy podłączyć urządzenie kablem USB. Część urządzeń trzeba programować za pomocą zewnętrznego konwertera.

Przed programowaniem modułów ESP należy doinstalować do Arduino IDE

- pakiet [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) umożliwiający programowanie modułów ESP8266 i
- pakiet [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json) umożliwiający programowanie modułów ESP32.

Wraz z pakietami instalowana jest bardzo duża ilość przykładowych programów, które obejmują podstawowe zastosowania modułów ESP. Dodatkowo można doinstalować biblioteki z dużą ilością kolejnych przykładowych programów.

W internecie znajduje się dużo witryn zawierających opis przykładowych programów wraz z instrukcją konfiguracji i realizacji połączeń do czujników i elementów wykonawczych. Bardzo użyteczną może okazać się witryna <https://randomnerdtutorials.com/> zawierająca wiele instrukcji filmowych i pisanych jak zrealizować i przetestować przykładowe programy. Autorzy witryny opublikowali też kilka płatnych podręczników.

### **Ćwiczenie 9.1 Instalacja środowiska programistycznego Arduino IDE, uruchomienie i testowanie płytki ESP8266.**

Polecenie dotyczy pracy zdalnej z domu, w laboratorium środowisko Arduino IDE jest zainstalowane a płytki są przetestowane.

1. Zainstalować Arduino IDE.
2. Podłączyć płytkę, z menu Narzędzia wybrać  
Narzędzia>Płytki>ESP8266 Boards(2.7.4)>NodeMCU 1.0 (ESP-12E Module) ....
3. Z menu Narzędzia>Port wybrać właściwy port COM, następnie otworzyć  
Narzędzia>Monitor portu szeregowego.
4. W monitorze portu szeregowego ustawić szybkość transmisji 115200.
5. Uruchomić i przetestować przykładowy program skanera sieci Wi-Fi  
Plik>Przykłady>ESP8266WiFi>WiFiScan.

Program WiFiScan jest programem, który często uruchamia się na nowych urządzeniach w celu ich przetestowania. Program nie wymaga żadnych zmian kodu i działa od razu po uruchomieniu. Program skanuje sieci Wi-Fi i wysyła komunikaty o wykryciu sieci na port szeregowy. Poniżej przykładowy efekt działania programu.

```
scan start  
scan done
```

Projekt „SezAM wiedzy, kompetencji i umiejętności” jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój

```
6 networks found
1: FunBox2-6FAF (-71) *
2: Porzycze (-71) *
3: AKA (-61) *
4: multimedia_grzegorz (-94) *
5: HomeNet (-84) *
6: HomeNet_Guest (-86) *
```

### Ćwiczenie 9.2. Program prostego serwera HTTP.

1. Wczytać przykładowy program prostego serwera HTTP  
Plik>Przykłady>ESP8266WebServer>HelloServer
2. W kodzie programu (szkicu) wprowadzić dane routera Wi-Fi

```
#ifndef STASSID
#define STASSID "your-ssid"
#define STAPSK "your-password"
#endif
```
3. Uruchomić program, następnie przetestować go z użyciem przeglądarki i programu Postman, numer IP serwera odczytać z monitora portu szeregowego, obsługiwane ścieżki i nazwę serwera można odczytać z kodu programu.

### Ćwiczenie 9.3. Program wykorzystujący RESTfull API do rejestracji wyników pomiarów.

1. Pobrać kod programu *PostHttpClient\_Do\_WebAPI.ino* przesyłającego do usługi WebAPI, opisanej w ćwiczeniu *Usługi internetowe RESTful WebAPI*, symulowane wyniki pomiarów
2. Wpisać w kodzie swoje dane (AUTHOR, GAGE) i dane swojej sieci Wi-Fi, następnie uruchomić program.
3. Zainstalować bibliotekę *Arduino\_JSON*.
4. Zaobserwować komunikaty wysyłane przez program na monitorze portu szeregowego.
5. Zaobserwować wysyłane przez program dane na stronie  
<http://argo.am.gdynia.pl/www/RestWyniki/Pages/WebForm1>

### Ćwiczenie 9.4. Program zdalnego sterowania pinami ESP8266 z wykorzystaniem protokołu MQTT

1. Podłączyć diody LED (światła drogowych) do pinów ESP8266: GND→G, R→D5, Y→D6, G→D7.
2. Zainstalować bibliotekę *PubSubClient*.
3. Pobrać kod programu *mqtt\_esp8266\_PINY.ino* zdalnego sterowania pinami ESP8266 z wykorzystaniem protokołu MQTT.

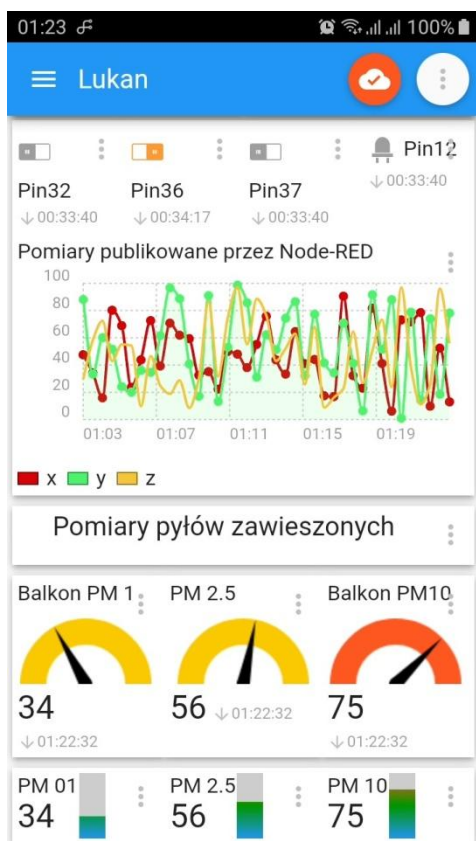
4. Wpisać w kodzie dane swojej sieci Wi-Fi.
5. Uruchomić program.
6. Za pomocą strony <http://lukan.sytes.net:1880/ui/> sterować pinami zaobserwować efekt, odczytać komunikaty o otrzymanych przez Moduł ESP8266 wiadomościach.

### **Ćwiczenie 9.5. Sterowanie pinami Moduł ESP8266 za pomocą programu działającego na urządzeniu przenośnym**

1. Zainstalować na urządzeniu przenośnym program IoT MQTT Panel  
<https://play.google.com/store/apps/details?id=snr.lab.iotmqttpanel.prod&hl=pl>.
2. Dodać połączenie do brokera `lukan.sytes.net` port domyślny 1883.
3. Utworzyć nowy pulpit (Dashboard).
4. Dodać i skonfigurować 3 kontrolki Switch do sterowania pinami Moduł ESP8266, nazwy tematów są widoczne w kodzie programu `mqtt_esp8266_PINY.ino` i na grafie <http://lukan.sytes.net:1880>.
5. Dodać i skonfigurować 3 kontrolki Gauge lub Vertical Meter do wyświetlania wyników pomiarów publikowanych w temacie `/TITI/pomiary/xyz1`. Wyniki są publikowane w formacie JSON np.: `{"x":36.017,"y":90.75,"z":39.52}`  
Aby z danych w formacie JSON pobrać składową `x` należy w polu `JasonPath for subscribe` wpisać `$.x`.
6. Dodać do pulpitu kontrolkę Line Graph i skonfigurować ją do wykreslania wyników pomiarów `x, y, z`.

Na rysunku 9.7 przedstawiono przykładowy wygląd skonfigurowanego pulpitu.





Rys. 9.7. Przykładowa konfiguracja pulpitu z ćwiczenia 9.5

### Projekt własnego systemu

1. Na grafie Node-RED dodać bloczek brokera MQTT aedes broker.
2. Stworzyć własny lokalny system wykorzystujący lokalnego brokera MQTT.

Projekt „SezAM wiedzy, kompetencji i umiejętności” jest współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego w ramach Programu Operacyjnego Wiedza Edukacja Rozwój