

Aktualności

Witamy w nowym roku szkolnym 2025/2026. Witamy w nowym numerze gazetki szkolnej ZSTiE. Witamy w nowej gazetce „TeleProjekt”. Celem niniejszego periodyku jest promocja wiedzy, nauki i zachęcenie zarówno uczniów jak i nauczycieli Zespołu Szkół Teleinformatycznych i Elektronicznych do podjęcia aktywności i czynnego udziału w projektach szkolnych i zawodowych.

W poniedziałek 1.IX odbyło się uroczyste rozpoczęcie roku szkolnego. Od teraz wracamy do trybu zadaniowego, skierowanego przede wszystkim na jeden cel: mistrzostwo w swoim zawodzie.



„TeleProjekt” to właśnie pomysł na implementację tego celu w świadomości nas wszystkich.

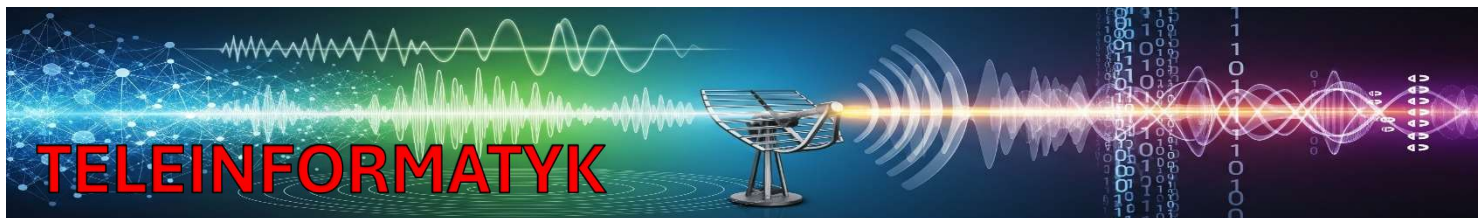
Spis Treści	
Aktualności.....	1
Zbuduj Własny Skaner Wi-Fi	2
Elektroniku - poznaj Fritzing	4
Arduino a sterowniki PLC	6
Krzyżówka	8

Wszystkich chętnych, rzecz jasna! Ruszmy z zapątem i wytrwajmy do końca naszej edukacji, która w naszych trzech zawodach trwać będzie dla każdego pewnie jeszcze długo po odebraniu dyplomu, świadectwa maturalnego czy życzeń powodzenia na Polibudzie / Uniwerku.

Dzisiejszy numer gazetki to artykuły kierunkowe związane z kierunkami ZSTiE, (Teleinformatyk, Elektronik i Robotyk) oraz krzyżówka. Od drugiego numeru będą dodatkowo publikowane egzaminacyjne testy zawodowe w formie **konkursu!** Jeżeli pomysł zostanie zaakceptowany i zyska uznanie Uczniów, zaprowadzimy listę rankingową i najlepsi „rozwiązywacze testów” będą mogli liczyć na docenienie ich zarówno przez nauczycieli, jak i organizatora tej małej inicjatywy. Tymczasem zachęcamy do lektury oraz kontaktu mailowego (haukeprojekt@gmail.com) w sprawach takich jak: propozycje tematów, pisanie artykułów, przesłanie rozwiązań testów i innych.

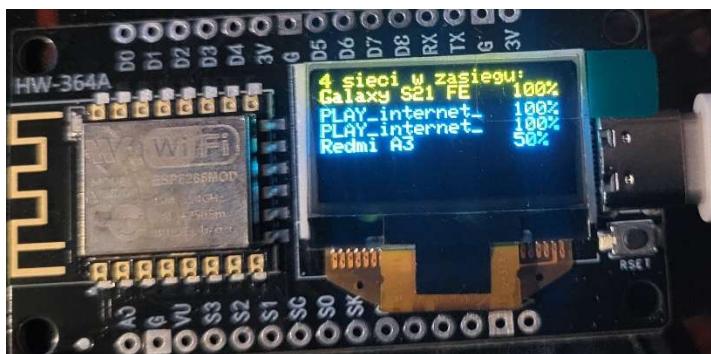
Autor tekstów jest nowym nauczycielem przedmiotów zawodowych na kierunku Robotyk. Absolwent Politechniki Wrocławskiej (AiR oraz EiT), nauczyciel w Zespole Szkół nr 2 (Borowska) w latach 2009-2025 na kierunkach samochodowych, konstruktor elektronik, pasjonat inżynierii wstecznej oprogramowania sterowników samochodowych, specjalista d.s. pojazdów elektrycznych w Laboratorium Badawczo-Rozwojowym Nysa Zakład Pojazdów S.A.

Stopka redakcyjna
 Redakcja: *Wojciech Błądek (sala 14 i 25)*



Zbuduj Własny Skaner Wi-Fi: Jak Zajrzeć w Niewidzialny Świat Fal Radiowych

Cześć, przyszli teleinformatycy! Codziennie łączymy się z Wi-Fi – w szkole, w domu, w autobusie. To dla nas tak naturalne jak oddychanie. Ale czy zastanawialiście się kiedyś, co tak naprawdę dzieje się w eterze? Ile sieci nas otacza i z jaką siłą "krzyczą" do naszych urządzeń? Dziś pokażemy, jak za pomocą prostego mikrokontrolera i małego wyświetlacza zbudować urządzenie, które odstani przed Wami ten niewidzialny świat.

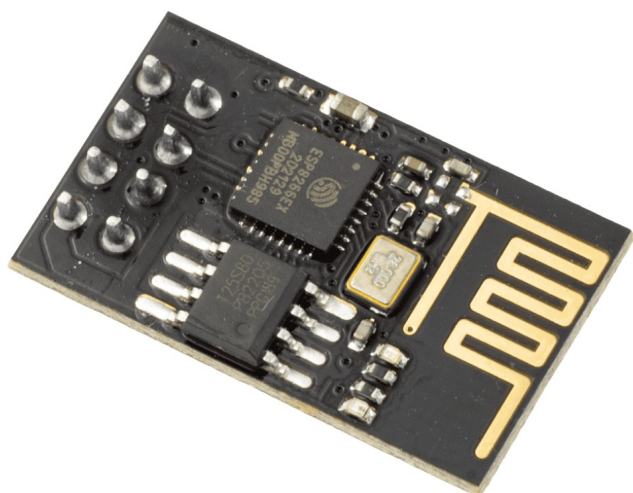


Rys. Na powyższym zdjęciu projekt zbudowany przez Autora

Sprzęt i Technologia: Serce Naszego Skanera

Nasz projekt opiera się na dwóch kluczowych elementach: mózgu, czyli mikrokontrolerze ESP8266, oraz oczach – wyświetlaczu OLED.

Mózg Operacji – czip ESP8266 to nie jest zwykły układ scalony. To potężny mikrokontroler z wbudowanym modułem Wi-Fi, który stał się legendą w świecie Internetu Rzeczy (IoT).



Jego największe zalety to niska cena, mały pobór mocy i ogromne możliwości. To dzięki niemu możemy nie tylko zasilać diodę, ale też łączyć się z globalną siecią, stawiać własne serwery czy, jak w naszym przypadku, nastuchiwać otaczających nas sieci.

Oczy, które Widzą Czerń: Wyświetlacz OLED W przeciwieństwie do standardowych ekranów LCD, które wymagają podświetlenia, wyświetlacze OLED (Organic Light Emitting Diode) same emitują światło. Każdy piksel jest małą, organiczną diodą. Co to oznacza w praktyce? Idealną czerń (bo piksele są po prostu wyłączone), nieskończony kontrast i niższy pobór energii. Idealny wybór do naszych przenośnych projektów!

Telekomunikacyjne Ciekawostki:

Magia Ukryta w Sygnale

Kiedy nasz skaner pokazuje siłę sygnału, nie wyświetla zwykłej liczby. W świecie telekomunikacji króluje RSSI (Received Signal Strength Indicator), czyli wskaźnik siły odbieranego sygnału. To tutaj zaczyna się prawdziwa magia:

- Skala w Minusie: Siłę sygnału mierzy się w decybelach-miliwatach (dBm). Jest to skala logarytmiczna, gdzie wartości są ujemne. Im wartość jest bliższa zera, tym sygnał jest silniejszy! Dlatego sygnał -50 dBm jest znacznie mocniejszy niż -80 dBm.
- Reguła 3 Decybeli: W skali logarytmicznej każda zmiana o 3 dB oznacza dwukrotną zmianę mocy. Oznacza to, że sygnał o sile -67 dBm jest dwa razy silniejszy niż sygnał o sile -70 dBm! Z kolei różnica 10 dB to już dziesięciokrotna różnica mocy.
- Walka o Kanał: Sieci Wi-Fi w paśmie 2.4 GHz działają na jednym z 11 lub 13 kanałów (w zależności od regionu). Działają trochę jak stacje radiowe – jeśli dwie sieci nadają na tym samym lub bliskim kanale, mogą się wzajemnie zakłócać.

Kod C++ dla środowiska Arduino IDE:

```
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// Definicja rozmiarów ekranu
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

// Definicja pinów I2C (tak jak w poprzednich przykładach)
#define OLED_SDA 14 // D6
#define OLED_SCL 12 // D5

// Utworzenie obiektu wyświetlacza
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire);

void setup() {
  // Inicjalizacja portu szeregowego do debugowania
  Serial.begin(115200);

  // Inicjalizacja wyświetlacza OLED
  Wire.begin(OLED_SDA, OLED_SCL);
  if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
    Serial.println(F("Błąd inicjalizacji SSD1306"));
    for(;;); // Zatrzymaj w razie błędu
  }

  // Ustawienie trybu Wi-Fi na "Station" (niezbędne do skanowania)
  WiFi.mode(WIFI_STA);
  // Rozłączenie z siecią, jeśli byliśmy połączeni
  WiFi.disconnect();
  delay(100);

  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(SSD1306_WHITE);
  display.setCursor(0, 0);
  display.println(F("Skaner WiFi gotowy"));
  display.display();
  delay(2000);
}

void loop() {
  scanAndDisplayNetworks();
  // Poczekaj 10 sekund przed kolejnym skanowaniem
  delay(10000);
}

// Funkcja do przeliczania RSSI na procenty
int getSignalStrengthAsPercent(int32_t rssi) {
  if (rssi <= -100) {
    return 0;
  } else if (rssi >= -50) {
    return 100;
  } else {
    // Używamy funkcji map do przeliczenia wartości z zakresu -
    100..-50 na 0..100
    return map(rssi, -100, -50, 0, 100);
  }
}

// Główna funkcja skanująca i wyświetlająca wyniki
void scanAndDisplayNetworks() {
  display.clearDisplay();
  display.setCursor(0, 0);
  display.println(F("Skanowanie sieci..."));
  display.display();

  // Rozpocznij skanowanie i pobierz liczbę znalezionych sieci
  int numNetworks = WiFi.scanNetworks();

  display.clearDisplay();

  if (numNetworks == 0) {
    display.setCursor(0, 0);
    display.println(F("Nie znaleziono sieci"));
  } else {
    display.setCursor(0, 0);
    display.print(numNetworks);
  }
}
```

```
display.println(F(" sieci w zasięgu:"));

// Wyświetl maksymalnie 7 sieci, aby zmieściły się na ekranie
int networksToDisplay = min(numNetworks, 7);

for (int i = 0; i < networksToDisplay; ++i) {
  // Pobierz nazwę sieci (SSID)
  String ssid = WiFi.SSID(i);
  // Pobierz siłę sygnału (RSSI)
  int32_t rssi = WiFi.RSSI(i);
  // Przelicz RSSI na procenty
  int percentage = getSignalStrengthAsPercent(rssi);

  // Przygotuj sformatowany tekst
  String line = ssid.substring(0, 14); // Skróć SSID, aby się
  zmieściło
  String signal = String(percentage) + "%";

  // Ustaw kursor dla nazwy sieci
  display.setCursor(0, (i * 8) + 8); // +8 pikseli na każdą
  linię
  display.print(line);

  // Ustaw kursor dla siły sygnału, wyrównując do prawej
  display.setCursor(100, (i * 8) + 8);
  display.print(signal);
}
// Wyślij wszystko na wyświetlacz
display.display();
}
```

Nasz Kod pod Lupą:

- **Wire.begin(...)** i **display.begin(...)**: Inicjalizacja komunikacji z wyświetlaczem OLED.
 - **WiFi.mode(WIFI_STA)**: To kluczowa komenda. Ustawia moduł Wi-Fi w tryb "Station", czyli klienta sieciowego. Tylko w tym trybie ESP8266 może aktywnie skanować otoczenie w poszukiwaniu punktów dostępowych.
 - **int numNetworks = WiFi.scanNetworks();**: To pojedyncza, ale potężna komenda. Zleca modułowi Wi-Fi przeskanowanie wszystkich kanałów i zwraca liczbę znalezionych sieci.
 - Pętla **for**: Po znalezieniu sieci, program w pętli pobiera informacje o każdej z nich: nazwę (SSID) oraz siłę sygnału (RSSI).
 - Funkcja **getSignalStrengthAsPercent()**: To nasz "tłumacz". Przekształca niezrozumiałe, ujemne wartości dBm (np. -65) na przyjazną dla użytkownika skalę procentową (0-100%), używając do tego prostej funkcji matematycznej **map()**.
- Podsumowując**, ten projekt jest idealnym przykładem teleinformatyki w praktyce – łączy w sobie programowanie mikrokontrolerów, zrozumienie działania sieci bezprzewodowych i wizualizację danych. To dowód na to, że za pomocą kilku linijek kodu i taniej elektroniki możemy budować niesamowite i użyteczne narzędzia.



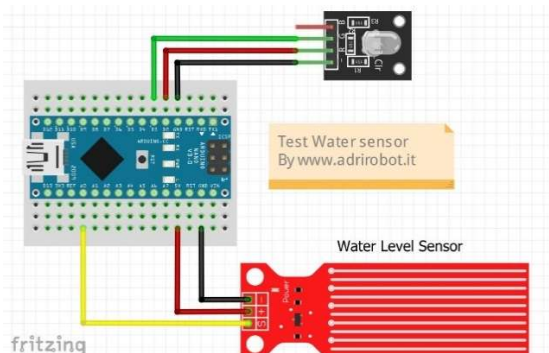
Od Chaosu na płytce do profesjonalnego projektu: poznaj Fritzing!

Cześć, młodzi elektronicy! Każdy z nas zna ten moment: godziny spędzone nad płytką stykową, plątanina kabli przypominająca spaghetti, a na końcu ta satysfakcja, gdy dioda wreszcie miga tak, jak powinna. Projekt działa! Ale co dalej? Jak zamienić ten prototyp w trwałe, profesjonalnie wyglądające urządzenie? Jak pokazać swój pomysł innym w zrozumiały sposób? Odpowiedzią na te pytania jest Fritzing – genialne oprogramowanie, które powinno znaleźć się w arsenale każdego adepta elektroniki.



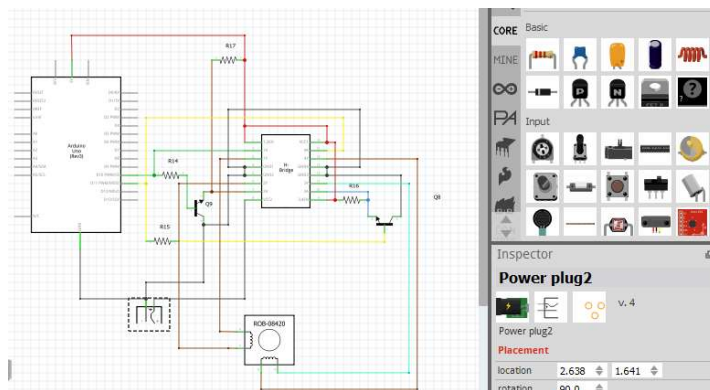
Co to jest Fritzing i dlaczego powinieneś go znać?

Fritzing to darmowe oprogramowanie typu open-source, stworzone z myślą o artystach, hobbystach i uczniach – czyli wszystkich tych, którzy chcą urzeczywistniać swoje elektroniczne pomysły bez konieczności opanowania skomplikowanych, inżynierskich narzędzi CAD. Jego największą siłą jest intuicyjność i praca w trzech zsynchronizowanych widokach, które prowadzą nas za rękę przez cały proces projektowy.

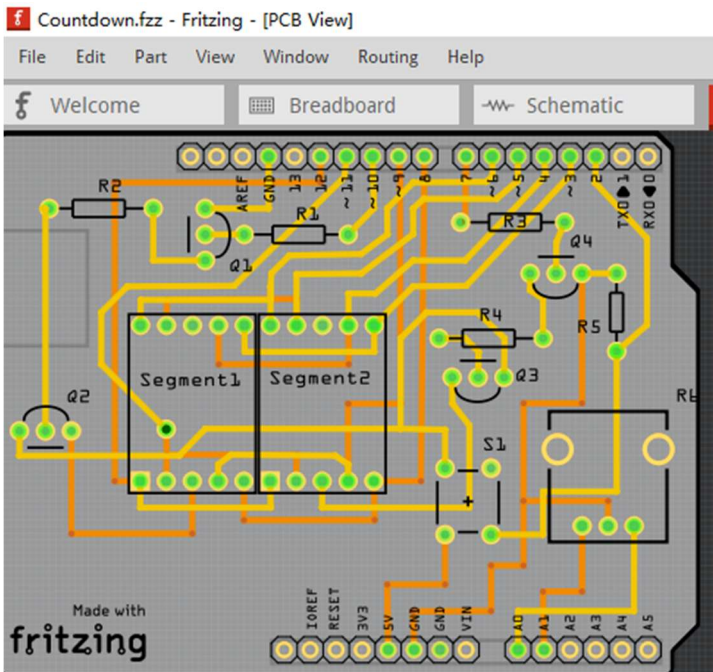


1. Widok Płytki Stykowej (Breadboard) To tutaj wszystko się zaczyna. Fritzing oferuje wirtualną płytkę stykową, która wygląda i działa dokładnie tak, jak ta na Twoim biurku. Przeciągasz komponenty z biblioteki, umieszczasz je w otworach i łączysz wirtualnymi kabelkami. To idealny sposób na zaplanowanie układu, zanim jeszcze wepniesz pierwszy rezystor. Koniec z bałaganem – teraz Twój prototyp jest czytelny i gotowy do udokumentowania!

2. Widok Schematu (Schematic) Gdy połączysz wszystko na płytce stykowej, Fritzing automatycznie generuje profesjonalny schemat ideowy! Program sam wie, które nóżki komponentów są ze sobą połączone i przedstawia to w formalnym języku elektroniki. To fantastyczne narzędzie do nauki, które pomaga zrozumieć teoretyczne podstawy działania układu, który właśnie zbudowałeś.



3. Widok Płytki Drukowanej (PCB) To tutaj Twój projekt nabiera realnych kształtów. Fritzing pozwala zaprojektować własną płytkę drukowaną (PCB), na której na stałe przylutujesz wszystkie elementy. Możesz dowolnie rozmieszczać komponenty i rysować ścieżki, a program dba o to, by połączenia zgadzały się ze schematem. To brama do tworzenia własnych, profesjonalnych urządzeń.



Pełen Arsenal Możliwości: Co jeszcze potrafi Fritzing?

Fritzing to znacznie więcej niż tylko ładne rysunki. To kompletne środowisko, które oferuje potężne możliwości:

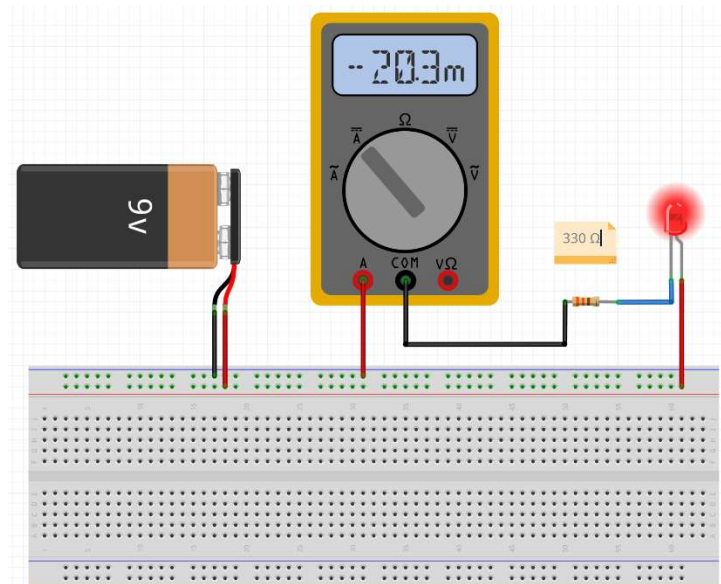
- **Od Projektu do Fabryki:** To najważniejsza funkcja dla zaawansowanych użytkowników. Fritzing potrafi wyeksportować Twój projekt PCB do plików **Gerber**. Jest to standardowy format przemysłowy, akceptowany przez fabryki na całym świecie. Oznacza to, że możesz wysłać swój plik np. do Chin i za kilka dolarów otrzymać profesjonalnie wykonane płytki drukowane prosto do domu!
- **Ogromna Biblioteka Komponentów:** Program posiada gigantyczną, stale rozwijaną bazę części – od podstawowych rezystorów i kondensatorów, przez układy scalone, aż po popularne moduły jak Arduino, Raspberry Pi czy setki czujników.
- **Tworzenie Własnych Elementów:** Brakuje jakiegoś egzotycznego komponentu w bibliotece? Żaden problem! Fritzing zawiera edytor, który pozwala na tworzenie własnych części i dodawanie ich do swoich projektów.
- **Inteligentny Pomocnik (Autoruter):** Jeśli nie chcesz ręcznie rysować wszystkich ścieżek na PCB, możesz skorzystać z funkcji autorutera. Jednym kliknięciem program sam spróbuje znaleźć najlepsze połączenia. To świetna pomoc na

początek, choć ręczne prowadzenie ścieżek daje często lepsze rezultaty.

- **Integracja z Kodem:** W specjalnym widoku "Kod" możesz pisać, edytować i przechowywać swój program (np. dla Arduino) bezpośrednio w pliku projektu. Dzięki temu cała dokumentacja – schemat, płytka i oprogramowanie – znajduje się w jednym miejscu.

Fritzing to Twój nowy najlepszy przyjaciel!

Fritzing to idealne narzędzie, które wypełnia lukę między teorią a praktyką. Pomaga unikać błędów, uczy czytania i tworzenia schematów, a co najważniejsze – obniża próg wejścia do świata projektowania własnych płytek PCB. To potężny kombajn, który zamieni Twoje pomysły w gotowe, profesjonalne urządzenia.



Fritzing to również pełnoprawny symulator obwodów elektrycznych. Komponenty mają zaimplementowaną fizykę i potrafią zadymić się przy przepaleniu, gałęzie wyliczyć prądy, napięcia i rezystancje, co jest ważne przy obwodach pomiarowych, jakże przydatnych w nauce elektroniki czy przygotowaniach na pracownię elektroniki!

Nie czekaj, pobierz go i wypróbuj przy następnym szkolnym projekcie. Gwarantuję, że już nigdy nie spojrzysz tak samo na swoją płytkę stykową!

Miłego projektowania!

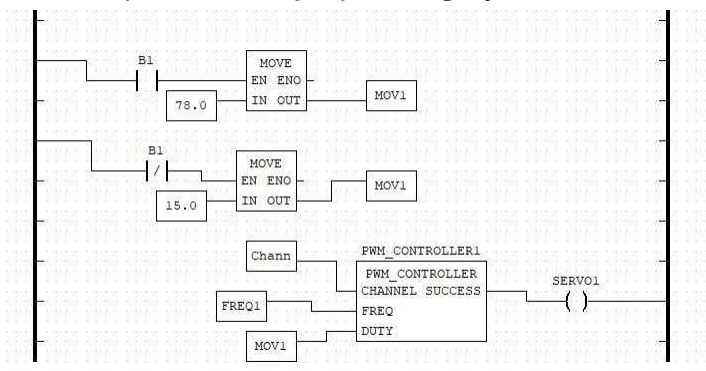
(WB)

nieskończonej pętli loop()). Sterowniki PLC działają inaczej, w oparciu o tzw. cykl skanu sterownika:

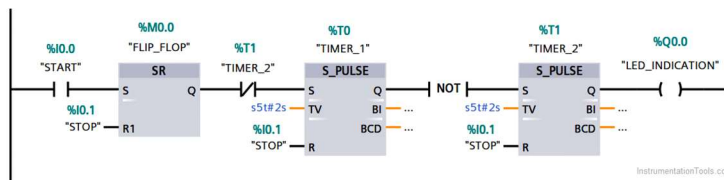
1. Odczytaj stan wszystkich wejść.
2. Wykonaj program (logikę).
3. Ustaw stany wszystkich wyjść.
4. Powtórz.

Ten cykliczny charakter pracy wymusił powstanie specyficznych języków, które na pierwszy rzut oka mogą wydawać się obce:

- LAD (Ladder Logic): To język graficzny, który wygląda jak schemat drabinkowy przekaźników elektrycznych. Został stworzony, aby elektrycy utrzymania ruchu mogli go łatwo zrozumieć. Zamiast pisać kod, "rysujesz" logikę.



- FBD (Function Block Diagram): Tutaj programujesz, łącząc gotowe bloki funkcyjne (AND, OR, timery, liczniki) w logiczną całość.



- SCL (Structured Control Language): To język tekstowy, podobny do Pascala. Jeśli programujesz w Arduino, SCL będzie dla Ciebie najbardziej naturalnym językiem w świecie PLC, ale nadal działa on w ramach cyklu skanu.

```
FUNCTION FC1: VOID
VAR
  Test1: STRUCT //ANY Struktur 1 anlegen
    ANY_id: BYTE; //Define ANY structure 1
    Source_DataType: BYTE;
    Source_Length: WORD;
    Source_DE_Nummer: WORD;
    Source_Byte_Pointer: DWORD;
  END_STRUCT;
  Test2: STRUCT //ANY Struktur 2 anlegen
    ANY_id: BYTE; //Define ANY structure 2
    Destin_DataType: BYTE;
    Destin_Length: WORD;
    Destin_DE_Nummer: WORD;
    Destin_Byte_Pointer: DWORD;
  END_STRUCT;
  //Deklaration ANY Pointer für Quell-DB
  //Declaring ANY pointer of source DB
  pAny_source RT Test1: ANY;
  //Deklaration ANY Pointer für Ziel-DB
  //Declaring ANY pointer of target DB
  pAny_destin RT Test2: ANY;
  erg: INT; //Rückgabewert / Return value
END_VAR
```

2. Niezawodność i środowisko pracy. Arduino jest stworzone do prototypowania. Siemens Simatic S7-1200 to "czołg" zaprojektowany do pracy 24/7 przez lata w trudnych warunkach przemysłowych (hałas elektryczny, wibracje, skrajne temperatury). Niezawodność jest tutaj absolutnym priorytetem.

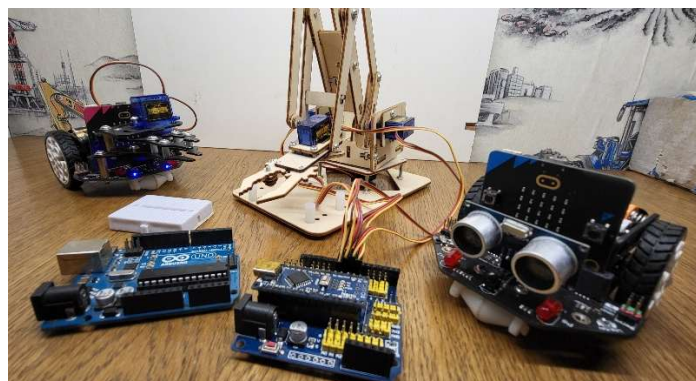
3. Zintegrowane Środowisko (TIA Portal). Arduino IDE to prosty edytor tekstu i kompilator. TIA (Totally Integrated Automation) Portal to potężny kombajn, w którym konfigurujesz nie tylko logikę sterownika, ale także panele HMI (interfejsy człowiek-maszyna), napędy, sieci przemysłowe i zaawansowaną diagnostykę.

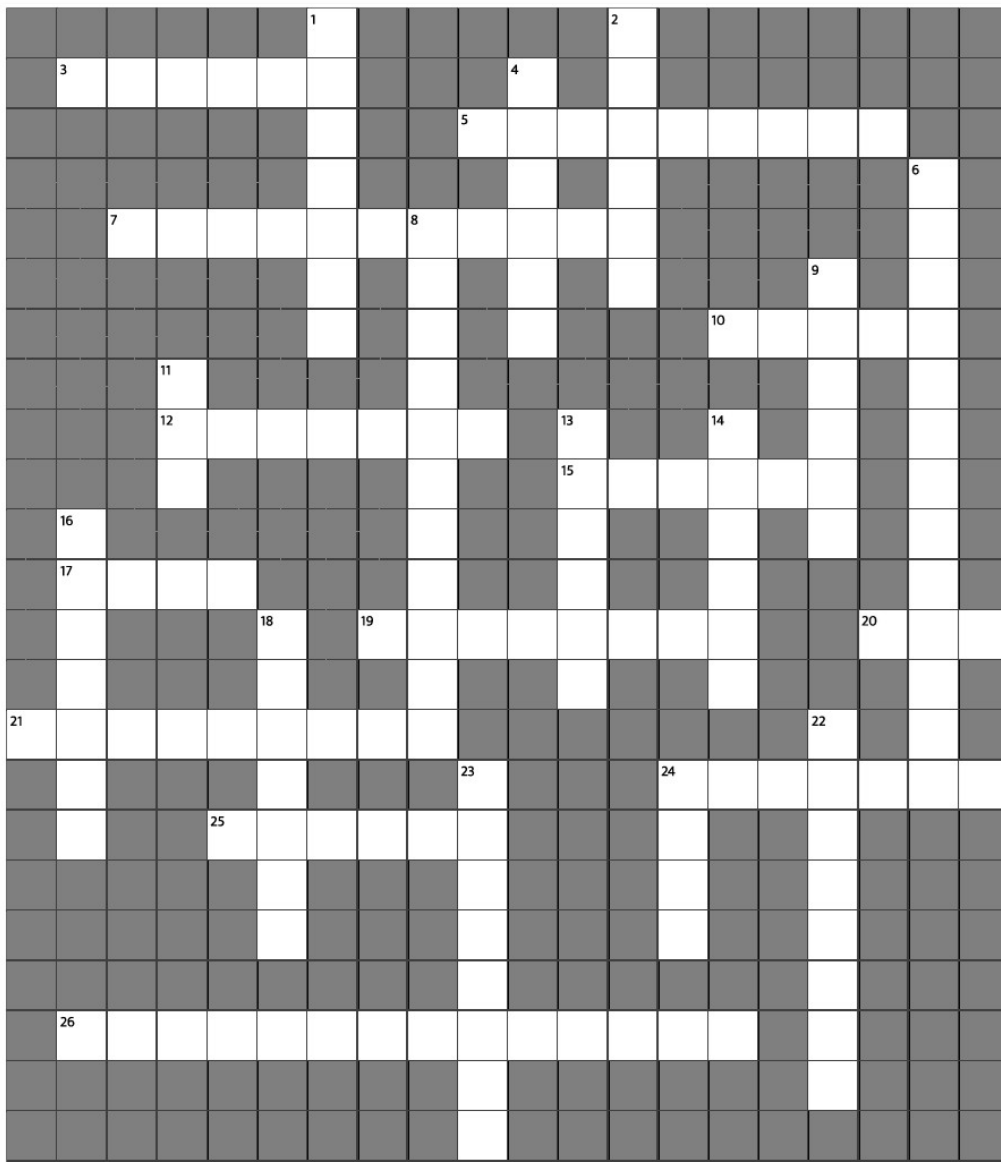
Werdykt: czy Arduino to dobry start?

Zdecydowanie tak! Arduino nie nauczy Cię składni języka LAD, ale da Ci coś znacznie cenniejszego: intuicję inżynierską i umiejętność logicznego myślenia.

Można to porównać do nauki jazdy. Jazda gokartem (Arduino) nie przygotuje Cię do prowadzenia 40-tonowej ciężarówki (PLC Siemens). Ale nauczy Cię podstaw: jak działa kierownica, gaz, hamulec i jak przewidywać tor jazdy. Gdy przesiądziesz się do ciężarówki, będziesz musiał nauczyć się obsługi skrzyni biegów i hamulców pneumatycznych (LAD, TIA Portal), ale fundamentalne zasady sterowania pojazdem już znasz.

Traktuj swoje projekty na Arduino jako trening dla umysłu. Kiedy w pracowni siadasz do TIA Portal, skup się na zrozumieniu nowej "skrzyni biegów" – cyklu skanu i graficznych języków programowania. Zobaczysz wtedy, że logika, którą chcesz zaimplementować, jest dokładnie taka sama, jakiej nauczyłeś się w domu. Twoje doświadczenie z Arduino da Ci ogromną przewagę, bo nie będziesz zaczynał od zera – będziesz już myślał jak automatyk.





K
R
Z
Y
Ż
Ó
W
K
A

poziomo

- 3** Uniwersalna podstawa programowania, wspólna dla Arduino i sterowników przemysłowych
- 5** Funkcja programu pozwalająca wirtualnie testować obwody i obserwować np. przepalenie elementu
- 7** Proces zabezpieczania transmisji danych w sieci, np. za pomocą protokołu WPA3
- 10** Półprzewodnikowy element, którego miganie jest oznaką sukcesu projektu
- 12** Elektroniczne "oko" lub "ucho" robota, które pozwala mu odbierać informacje z otoczenia
- 15** Element routera odpowiedzialny za nadawanie i odbieranie fal radiowych
- 17** wskaźnik siły odbieranego sygnału
- 19** Opis: Jedna z podstawowych wielkości fizycznych, którą potrafi obliczyć program w trybie symulacji
- 20** Tekstowy język programowania PLC, podobny do Pascala i naturalny dla osób znających C++
- 21** jednostka pasma WiFi
- 24** Formalny, ideowy rysunek połączeń, który Fritzing generuje automatycznie
- 25** Potężny komputer w sieci, który przechowuje strony internetowe lub udostępnia usługi
- 26** Układ scalony, na przykład ESP8266

w dół

- 1** Przenośne źródło zasilania, niezbędne dla mobilnych robotów
- 2** Urządzenie tworzące lokalną sieć bezprzewodową i udostępniające internet
- 4** Może być cyfrowy lub analogowy; przenosi informacje w sieciach i układach elektronicznych
- 6** skala dBm-ów
- 8** OLED lub LCD
- 9** TIA, nie tylko w Minecraftcie
- 11** Skrót od nazwy płytki drukowanej, na której na stałe montuje się komponenty
- 13** Miejsce, w którym komputer, telefon lub robot przechowuje dane i programy
- 14** Przemysłowy format plików, do którego eksportuje się projekt w celu wysłania go do fabryki
- 16** Popularny moduł z mikrokontrolerem, dostępny w bibliotece komponentów Fritzing
- 18** Producent profesjonalnych sterowników PLC, np. S7 1200
- 22** Podstawowy komponent elektroniczny, o którym mowa przy pierwszym wpinaniu w płytkę
- 23** Pierwsza, testowa wersja układu, często budowana na płytce stykowej
- 24** Unikalna nazwa, która identyfikuje sieć bezprzewodową