

TABLICA EDUKACYJNA

A1

STARTER KIT

Podstawy nauki programowania
mikrokontrolerów
i zaznajomienie z obwodami
elektronicznymi

AtMega 2560

SPIS TREŚCI

1 Specyfikacja i opis tablicy edukacyjnej A1	- 4
1.1 Informacje ogólne	- 4
1.2 Wykaz elementów składowych tablicy A1	- 4
1.3 Ogólne zasady bezpieczeństwa korzystania z tablicy A1	- 4
1.4 Rozmieszczenie elementów na płycie czołowej	- 5
2 Uruchomienie	- 5
3 Podstawowe zagadnienia z teorii obwodów elektrycznych	- 6
3.1 Wspólna masa układu w tablicy A1	- 6
3.2 Układ wzmacniający z tranzystorem	- 6
3.3 Diody	- 7
3.4 Przykładowy obwód elektryczny z diodą LED	- 7
3.5 Działanie czujnika PIR	- 8
3.6 Działanie czujnika zbliżeniowego	- 9
3.7 Działanie czujnika elektrochemicznego	- 9
4 Mikrokontroler informacje wprowadzające	- 10
4.1 Co to jest mikrokontroler	- 10
4.2 Główne cechy mikrokontrolera AtMega2560	- 10
5 Instalacja środowiska programistycznego Arduino IDE	- 11
5.1 Pobieranie i instalowanie oprogramowania	- 12
6 Lekcje programowania	- 12
6.1 Sterowanie diodami	- 12
6.2 Instrukcja pinMode	- 12
6.3 Instrukcja digitalWrite	- 12
6.4 Instrukcja delay	- 12
6.5 Lekcja 2	- 13
6.6 Instrukcja define	- 14
6.7 Instrukcja digitalRead	- 14
6.8 Pętla for	- 14

1. Specyfikacja i opis tablicy edukacyjnej A1

1.1 Informacje ogólne

Tablica jest zespolonym urządzeniem demonstracyjnym umożliwiającym samodzielne zestawianie połączeń pomiędzy mikrokontrolerem a różnorodnymi elementami wykonawczymi, czujnikami, potencjometrem i przyciskami.

Tablica ma pomóc zrozumieć ogólną istotę funkcjonowania układów elektronicznych, obwodów elektrycznych a także działania mikrokontrolera i zasady jego programowania.

1.2 Wykaz elementów składowych tablicy A1

1. Płyta PCB zgodna ze specyfikacją Arduino Mega z mikrokontrolerem AtMega 2560
 2. Gniazda bananowe x 41
 4. Przewody różnokolorowe obustronnie zakończone wtykiem bananowym x 10
 5. Przyciski monostabilne x 3
 6. Woltomierz cyfrowy o zakresie od 0V do 30 V
 7. Żarówka E12 sterowana poprzez układ tranzystorowy z tranzystorem BD135 i rezystorem 1kOhm
 8. Włącznik główny
 9. Włącznik zasilania silnika krokowego
 10. Włącznik zasilania serwo
 11. Diody LED sygnalizujące zasilanie silnika krokowego i serwo
 12. Diody LED z rezystorami zabezpieczającymi o kolorach CZERWONA, ŻÓŁTA, ZIELONA, NIEBIESKA, BIAŁA
 13. Wyświetlacz TFT LCD1602 ze sterownikiem HD44780
 14. Głośnik 0,25W / 8 Ohm z rezystorem zabezpieczającym 47 Ohm
 15. Czujnik tlenku węgla MQ-3
 16. Czujnik ruchu w podczerwieni
 17. Diodowy czujnik zbliżeniowy
 18. Ultradźwiękowy czujnik odległości
 19. Zintegrowany cyfrowy czujnik temperatury i wilgotności DHT 22
 20. Potencjometr obrotowy 10kOhm z gałką
 21. Czujnik Halla (hallotron - detektor pola magnetycznego)
 22. Fotorezystor 10kOhm
 23. Termistor 10kOhm
 24. Silnik krokowy NEMA17 ze sterownikiem STEPSTICK
 25. Micro-serwo
 26. Zespół mikroprzełączników 2 pozycyjnych tzw dipswitch
 27. Przewód USB A-B
- Elementy niedostępne dla użytkownika:
28. Stabilizowany zasilacz sieciowy 230/12V 2A z zabezpieczeniem przeciwzwarciovym
 29. Układ obniżenia napięcia step-down 2A 12->5 V z zabezpieczeniem przeciwzwarciovym

Rozmieszczenie wymienionych elementów pokazano na **ilustracji nr 1**

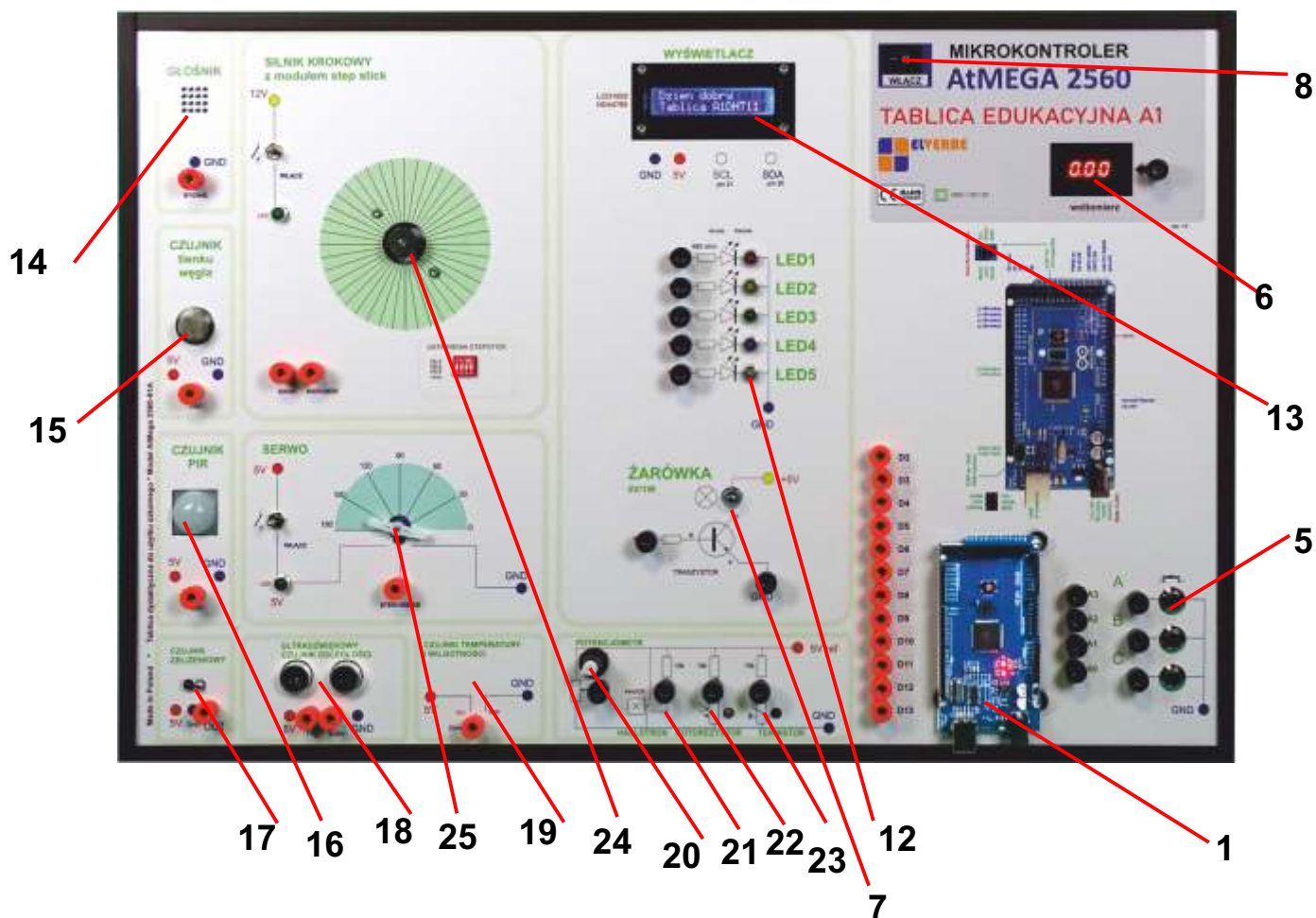
1.3 Ogólne zasady bezpieczeństwa korzystania z Tablicy edukacyjnej A1

Mimo, iż konstrukcja tablicy zapewnia całkowite bezpieczeństwo użytkowania, obudowa jest dielektrykiem (materiałem nieprzewodzącym prąd), a wszystkie dostępne elementy mogą znajdować się pod napięciem nie wyższym niż 5 lub 12 voltów, zaleca się zaznajomienie uczniów z ogólnymi zasadami bezpieczeństwa w obchodzeniu się z urządzeniami elektronicznymi.

1. Urządzenie można podłączać wyłącznie do sprawnie działającej sieci energetycznej 230V.
2. Nie należy włączać tablicy w przypadku stwierdzenia przetarcia kabla sieciowego, uszkodzonej wtyczki lub bolców sieciowych.
3. Podłoga w sali lekcyjnej lub pomieszczeniu, w którym tablica będzie użytkowana powinna być izolowana, nie może przewodzić prądu, być mokra itp.
4. W trakcie zajęć i korzystania z tablicy zabrania się spożywania napojów, jedzenia, użytkownicy powinni pracować suchymi rękoma, na przegubach dłoni nie mogą znajdować się metalowe łańcuszki.
5. Połączenia pomiędzy elementami można wykonywać wyłącznie z zastosowaniem dołączonych do zestawu przewodów z wtykami bananowymi. Konstrukcja tablicy zapewnia bezawaryjną pracę i brak możliwości uszkodzenia tablicy nawet jeśli połączenia bananowe wydawać się będą nielogiczne. Żadne z możliwych połączeń nie może spowodować zwarcia elektrycznego.

1. Specyfikacja i opis tablicy edukacyjnej A1

1.4 Ilustracja nr 1 Rozmieszczenie elementów na płycie czołowej



2. Uruchomienie

2.1 Tablica może być uruchamiana w pozycji poziomej (leżącej), pionowo (na stole) albo pionowo wieszana na ścianie. W przypadku wieszania na ścianie należy przykręcić do tylnej obudowy uchwyty mocujące do powieszenia (nie znajdują się w zestawie).

2.2 W celu uruchomienia należy podłączyć tablicę do sieci 230V i ustawić włącznik główny (8) w pozycję 1. Zasilanie silnika krokowego jak i zasilanie serwo powinno być wyłączone (przełączniki w pozycji do góry - diody sygnalizacyjne zasilanie nie powinny się palić).

Po włączeniu zasilania powinna zapalić się dioda sygnalizacyjna na płycie PCB mikrokontrolera, uruchomić powinien się woltomierz i wskazywać napięcie 0.00 V oraz uruchomić się wyświetlacz LCD.

Stan ten prezentuje ilustracja nr 2 - Typowe zachowanie tablicy po pierwszym włączeniu.

Uwaga: W zależności od tego jaki program startowy został zainstalowany w mikrokontrolerze, wyświetlacz może pokazywać inne treści. Prawidłową pracę wskazuje fakt podświetlenia wyświetlacza i pojawiający się napis powitalny.

Zanim nauczyciel rozpocznie zajęcia prezentujące działanie mikrokontrolera wskazane jest, aby uczniowie zapoznali się z podstawami teorii obwodów elektrycznych - najważniejsze zagadnienia w tym zakresie opisano w rozdziale 3. Zakłada się, że nauczyciel posiada wiedzę z zakresu podstaw obwodów.

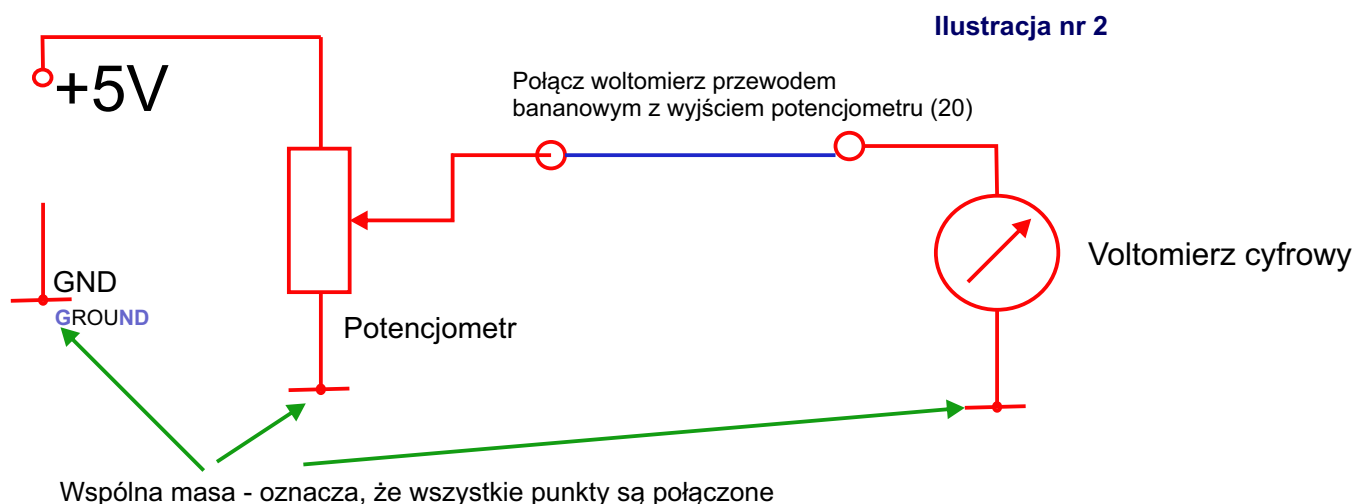
3. Podstawowe zagadnienia z teorii obwodów elektrycznych

3.1 Wspólna masa układu w Tablicy A1

Tablica została wykonana w taki sposób, aby uczniowie nie byli w stanie poprzez błędne podłączenie zasilania uszkodzić jakiegokolwiek element. Oznacza to, że zasilanie mikrokontrolera jak i elementów sterujących, czujników, silnika i serwo zostało już podłączone.

Ujemne bieguny diod (katody) oraz pozostałych elementów, które wymagają połączenia z ujemnym biegunem zasilania zostały połączone w tzw. wspólną masę.

Poniższy schemat (ilustracja nr 2) pokazuje w jaki sposób woltomierzem można zmierzyć napięcie w dowolnym punkcie Tablicy - tu na przykładzie napięcia na wyjściu potencjometru.



W tym doświadczeniu wyjaśniamy działanie potencjometru i działanie woltomierza.

Kręćąc gałką potencjometru jego suwak może połączyć się z punktem o najniższym potencjale (minimum potencjometru) - wówczas woltomierz wskazuje najniższe możliwe napięcie (czyli 0.00 V).

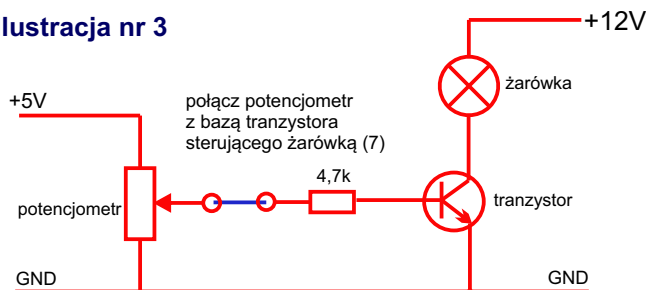
Przekręcając gałkę w drugą stronę łączymy suwak potencjometru z punktem o najwyższym możliwym potencjale.

3.2 Układ wzmacniający z tranzystorem

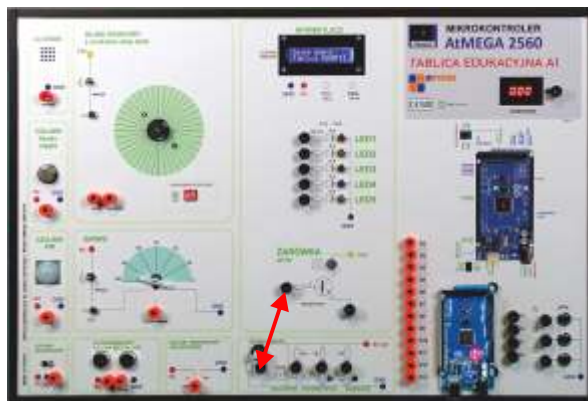
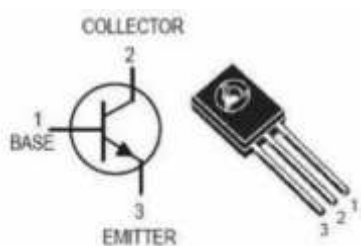
Tranzystor jest elektronicznym elementem półprzewodnikowym o trzech wyprowadzeniach oznaczonych jako BAZA, KOLEKTOR i EMITER. Jeżeli BAZA tranzystora nie jest odpowiednio spolaryzowana (nie ma do niej przyłożonego napięcia, czyli przez złącze BAZA-EMITER nie płynie prąd) to tranzystor nie przewodzi prądu.

Taka sytuacja występuje po włączeniu Tablicy. W obwód tranzystora włączona jest żarówka (7). Jeżeli do BAZY tranzystora podamy napięcie (np. z potencjometru) możemy regulować przepływ prądu w obwodzie KOLEKTOR - EMITER, a tym samym sterować jasnością świecenia żarówki.

Ilustracja nr 3



Ilustracja nr 4



3. Podstawowe zagadnienia z teorii obwodów elektrycznych

3.3 Diody

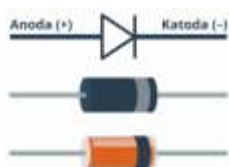
Dioda to element półprzewodnikowy posiadający dwa wyprowadzenia (złącza). Nazywamy je anodą i katodą. Główną właściwością diody jest przewodnictwo jednokierunkowe.

Dioda przewodzi prąd tylko wówczas, gdy jej złącze zwane anodą podłączone jest do dodatniego bieguna zasilania, zaś katoda do ujemnego (na tablicy oznaczone GND - od angielskiego słowa GROUND).

Diody elektroluminescyjne (LED) wraz z przewodzeniem prądu emitują światło. W zależności od zastosowanego materiału półprzewodnikowego i jego domieszek uzyskuje się konkretną barwę światła (długość fali). Dzięki temu istnieją diody świecące światłem podczerwonym, widzialnym albo ultrafioletowym. Na Tablicy A1 zastosowane są diody LED o 5 różnych barwach, które będziemy wykorzystywać w naszych projektach.

W celu zabezpieczenia diody przed nadmiernym prądem, który mógłby spowodować przepalenie, szeregowo z każdą diodą połączony jest rezystor zabezpieczający o wartości 760 Ohm.

Ilustracja 5. Symbol diody

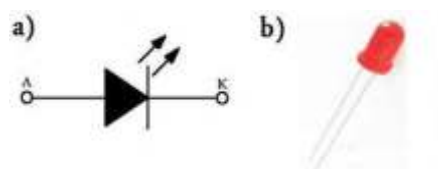


Przykładowy wygląd diod
(prostownicza, niżej impulsowa)

Ilustracja 6.

a) Symbol diody LED

b) Wygląd diody LED czerwonej



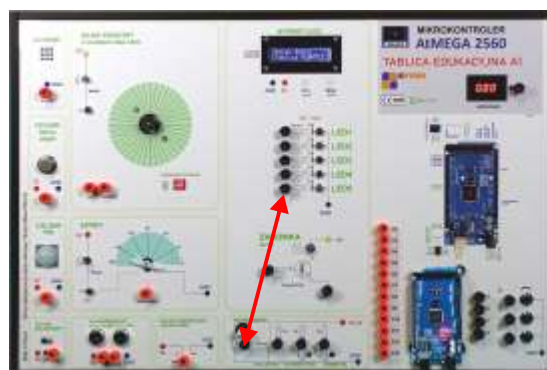
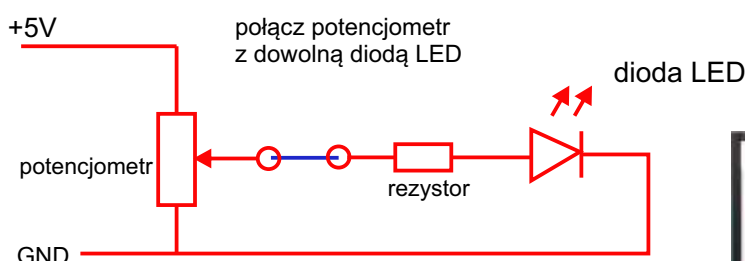
3.4 Przykładowy obwód elektryczny z diodą LED

Wykonajmy połączenie przewodem bananowym dowolnej diody LED (12) z potencjometrem

Uzyskamy obwód elektryczny jak na ilustracji 7

Obracając gałkę potencjometru regulujemy wartość napięcia, które przewodem bananowym przenosimy na anodę diody LED. Przy pewnym napięciu (zwykle wynosi ono od 1.3V do 1.9V zależnie od typu diody) zaczyna ona świecić. Regulując napięcie potencjometrem uzyskujemy inną jasność diody.

Ilustracja nr 7 Połączenie diody LED z potencjometrem



3. Podstawowe zagadnienia z teorii obwodów elektrycznych

3.5 Działanie czujnika ruchu (PIR)

PIR (ang. Passive Infra Red - pasywny czujnik podczerwieni) - czujnik elektroniczny służący do wykrywania ruchu. Powszechnie stosowany w systemach alarmowych, systemach automatycznego załączania oświetlenia i wentylacji itp.

Czujnik na Tablicy to element nr 16. Posiada on 3 złącza (wyprowadzenia), dwa z nich to zasilanie (+5V i GND) trzeci to wyjście sygnału. Gdy czujnik nie wykrywa ruchu na wyjściu czujnika możemy zmierzyć napięcie 5V (stan wysoki), a w chwili wykrycia ruchu - zmierzmy napięcie 0V (stan niski).

Gnizado 16 - czujnika ruchu możemy podłączyć do woltomierza i zmierzyć napięcie

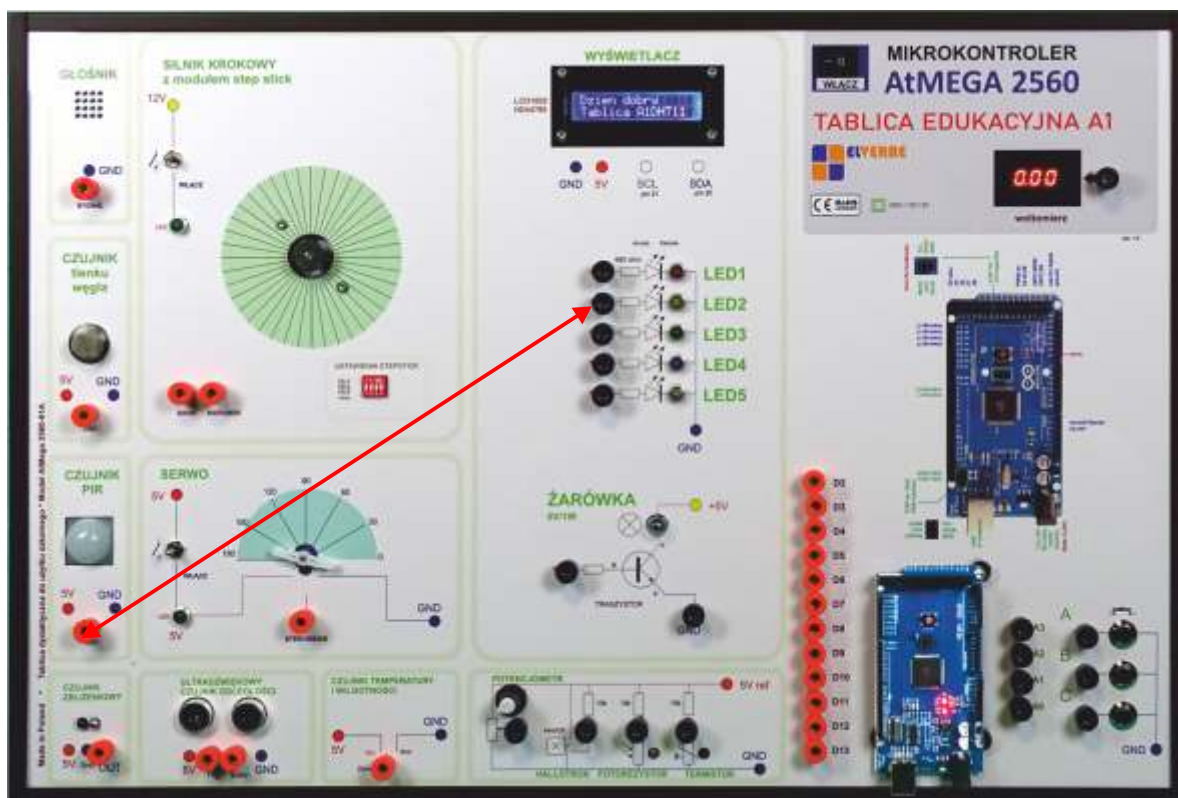
Gdy czujnik ruchu wykryje ruch woltomierz wskaże 0V.

Odsuń się od tablicy - odczekaj chwilę bez ruchu. Woltomierz po krótkim czasie wskaże napięcie 5V, ale gdy tylko się poruszyś pokaże znowu 0V



Ilustracja nr 8 Połączenie czujnika PIR z woltomierzem

Ilustracja nr 9. Zamiast do woltomierza możemy połączyć czujnik PIR z dowolną diodą LED



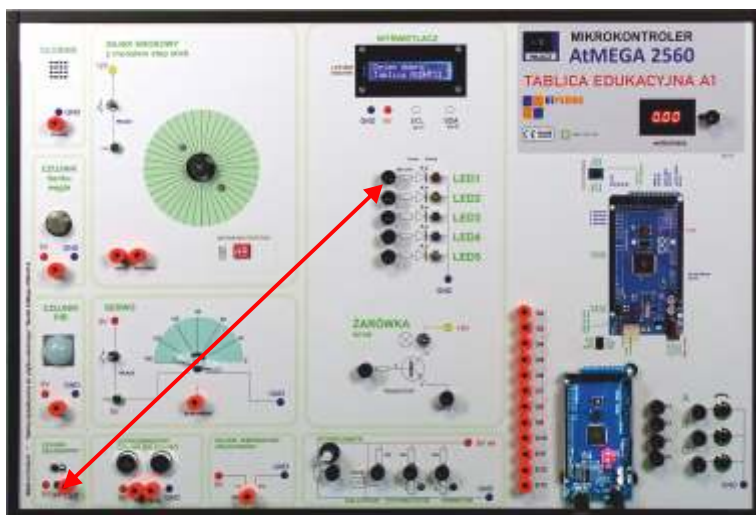
3. Podstawowe zagadnienia z teorii obwodów elektrycznych

3.6 Działanie czujnika zbliżeniowego

Czujnik zbliżeniowy zbudowany jest z dwóch diod. Pierwsza z nich emituje światło w podczerwieni (dioda LED infrared - dioda nadawcza), a druga dioda umieszczona obok - to fotodioda, która jest wrażliwa na światło podczerwone. Jeżeli tuż przed diodami pojawi się przedmiot odbijający światło, dioda odbiorcza zacznie przewodzić prąd.

Wykonajmy połączenie czujnika zbliżeniowego (17) z woltomierzem a później z dowolną diodą LED. Zbliżajmy małą, białą kartkę papieru przed czujnikiem zbliżeniowym. Obserwujemy wskazanie woltomierza albo diody LED.

Ilustracja nr 10 Połączenie czujnika zbliżeniowego z diodą LED.



3.7 Działanie czujnika elektrochemicznego

Czujnik elektrochemiczny - to element półprzewodnikowy wrażliwy na konkretny związek chemiczny, albo też cząstki określonego gazu. Zastosowany na Tablicy A1 czujnik jest wrażliwy na cząstki tlenku węgla czyli związku o chemicznym wzorze **CO**

Tlenek węgla zwany też czadem jest gazem niebezpiecznym, który w skrajnych przypadkach może prowadzić do śmierci. Z tego powodu czujniki alarmowe czadu są urządzeniami powszechnie stosowanymi w instalacjach alarmowych p-poż, ale też w domowych czujnikach bezpieczeństwa.

Zastosowany czujnik pozwoli nam przeprowadzić wiele różnych eksperymentów, jest on bardzo czuły nawet na tak małe stężenia tlenku węgla, które dla człowieka nie są groźne, ani tym bardziej wyczuwalne (czad jako związek nie jest wyczuwalny nawet w dużych, niebezpiecznych stężeniach).

Aby sprawdzić działanie czujnika podłączymy wyjście czujnika z woltomierzem. Do tego doświadczenia przyda się dowolna pusta puszka po napoju (np coca-coli), do której nauczyciel może wrzucać palące się zapalki. Zapalki gasnąc w puszcze wydzielają charakterystyczny zapach palonego drewna, ale już po wrzuceniu 3-4 zapalek wydobywający się „dymek” z puszki możemy zbliżyć do czujnika czadu i obserwujemy po kilku sekundach odczyt woltomierza. Wartość ta wyraźnie się zmieni co oznacza wykrycie tlenku węgla wydobywającego się z puszki. Z pewnością ilość czadu nie będzie szkodliwa dla człowieka, jednak wykrywalna przez czujnik.

Opisany eksperyment może wykonać tylko nauczyciel. Uczniowie jak i nauczyciel nie powinni wdychać oparów wydobywających się z puszki przeznaczonej do eksperymentu. Należy też zachować szczególną ostrożność w chwili podpalania zapalki i wrzucania jej do puszki, zważając aby nie zaproszyć ognia!

4. Mikrokontroler informacje wprowadzające

4.1 Co to jest mikrokontroler?

Mikrokontroler to użyteczny i całkowicie autonomiczny system mikroprocesorowy, niewymagający użycia dodatkowych elementów, których wymagałby do pracy tradycyjny mikroprocesor. Nazwać go można także mikrokomputerem jednoukładowym, ponieważ w jednym układzie scalonym posiada mikroprocesor, pamięć RAM oraz rozbudowane układy wejścia-wyjścia.

Zastosowany w tablicy mikrokontroler to bardzo rozpowszechniony układ oparty o mikrochip firmy ATMEL o symbolu AtMega2560.

4.2 Główne cechy Arduino Mega 2560:

Zainstalowany bootloader - dzięki zainstalowanemu bootloaderowi do zaprogramowania urządzenia wystarczy odpowiedni przewód USB oraz oprogramowanie ze strony producenta.

Arduino Shield - Rozkład złącz umożliwia montaż dedykowanych nakładek tzw. Arduino Shield.

54 cyfrowych wejść/wyjść umożliwia m.in. sterowanie diodami LED, przekaźnikami oraz odczytywanie stanów przycisków.

Maksymalna wydajność prądowa pojedynczego wyprowadzenia wynosi 40 mA.

15 wyjść PWM pozwala np. na sterowanie silnikami oraz regulowanie jasności diod.

16 wejść wbudowanego przetwornika analogowo-cyfrowego o rozdzielczości 10-bitów obsługuje m.in. czujniki z wyjściem analogowym

Komunikacja szeregową - Urządzenie obsługuje popularne interfejsy komunikacyjne, m.in.: UART, I2C i SPI.

Pamięć wbudowana - Układ ATmega2560 taktowany jest sygnałem o częstotliwości 16 MHz, posiada 256 kB pamięci programu Flash, 4 kB EEPROM oraz 8 kB pamięci operacyjnej SRAM.

Do zasilania Arduino można wykorzystać dowolny zasilacz o napięciu od 7V do 12V ze złączem DC 5,5 x 2,1 mm.

Płytkę można zasiląć również z komputera poprzez przewód USB pamiętając przy tym, że maksymalna wydajność prądowa portu USB wynosi 500 mA. Arduino posiada system chroniący gniazdo przed zwarcieniem oraz przepływem zbyt wysokiego prądu.

Złącze ICSP - Moduł posiada wyprowadzenia ICSP służące do podłączenia zewnętrznego programatora AVR. Pin IOREF - Pin IOREF umożliwia bezpośredni dostęp do napięcia z jakim pracują wyprowadzenia wejść/wyjść.

Wbudowana dioda LED - Podłączona dioda LED na pinie 13 umożliwia debuggowanie prostych programów.

Wyjście 3,3 V - Wbudowany regulator napięcia umożliwia zasilanie zewnętrznych urządzeń napięciem 3,3 V o poborze prądu do 50 mA.

Specyfikacja produktu - dane charakterystyczne

Napięcie zasilania: 12 V

Mikrokontroler: ATmega 2560

- Maksymalna częstotliwość zegara: 16 MHz

- Pamięć SRAM: 8 kB

- Pamięć Flash: 256 kB (8 kB zarezerwowane dla bootloadera)

- Pamięć EEPROM: 4 kB

Piny I/O: 54

kanaly PWM: 15

Ilość wejść analogowych: 16 (kanały przetwornika A/C o rozdzielczości 10 bitów)

Interfejsy szeregowo: 4xUART, SPI, I2C

Zewnętrzne przerwania

Podłączona dioda LED do pinu 13

Gniazdo USB A do programowania

Złącze DC 5,5 x 2,1 mm do zasilania płytki



5. Mikrokontroler instalacja środowiska programistycznego

5.1 Pobieranie oprogramowania

Wszystkie projekty oparte będą o pracę w środowisku ARDUINO IDE - a podane tu przykłady oparte o system operacyjny Windows 10. Istnieje jednak możliwość zainstalowania środowiska ARDUINO IDE także na innym systemach operacyjnych takich jak Linux, iOS, a nawet Chromebox. Alternatywą jest też praca w chmurze bez potrzeby instalacji oprogramowania, ale wymaga to założenia konta i profilu na portalu <https://arduino.cc>

Wejdź w zakładkę **PROGRAMY NA MIKROKONTROLERY**

Pobierz i zainstaluj program ARDUINO IDE

Możesz pobrać to środowisko programistyczne bezpośrednio ze strony producenta:

www.arduino.cc

Zastosowane w Tablicy elementy zewnętrzne do poprawnej pracy wymagają użycia dodatkowych bibliotek programowych. Wszystkie dostępne są na stronie: <https://laboratoria-przyszlosci.edu.pl>

Pobierz biblioteki - będą to osobne foldery, które po ściągnięciu na komputer umieścić w katalogu: dokumenty/arduino/libraries

Teraz możesz uruchomić środowisko programistyczne ARDUINO IDE

Darmowe przykładowe programy możesz pobrać z naszej strony:

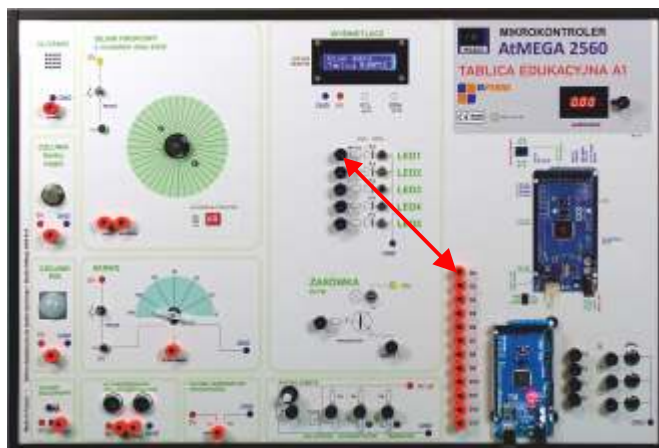
<https://laboratoria-przyszlosci.edu.pl> z zakładki **PROGRAMY NA MIKROKONTROLERY**

6 Lekcje programowania

6.1 LEKCJA 1: Sterowanie diodami (migające światelko)

Połącz przewodem bananowym pin D2 mikrokontrolera z diodą LED

Uwaga: nie ma znaczenia którą diodę wybierzesz - możesz w trakcie działania programu odpinąć przewód od tej diody i wpinać do innej diody a nawet do żarówki - efekt zawsze będzie taki sam - możesz programowo sterować świeceniem wybranej diody lub żarówki.



```
void setup() {  
  pinMode(2, OUTPUT);  
}
```

```
void loop() {  
  digitalWrite(2,1);  
  delay(500);  
  digitalWrite(2,0);  
  delay(500);  
}
```

6.2 funkcje wbudowane:

**void()
loop()**

Funkcje pozwalają na tworzenie funkcjonalnych fragmentów kodu. Może je dowolnie tworzyć użytkownik (programista), ale dwie funkcje - tzn void i loop - to wbudowane funkcje, które nie zwracają żadnych wartości, a służą do przeprowadzenia rozruchu (czyli „setup”) oraz do wykonywania poleceń w nieskończonej pętli („loop”);

W funkcji „setup” umieszczamy polecenia, którymi ustawiamy np. początkowe tryby pracy pinów mikrokontrolera.

6. Lekcje programowania

6.2 INSTRUKCJA `pinMode`

Ta instrukcja powoduje ustawienie konkretnego pinu mikrokontrolera do pracy jako wejście, albo wyjście sygnału. Należy zwrócić uwagę na pisownię instrukcji - pierwsza litera „p” pisana z małej litery, natomiast **Mode** z dużej (nie ma spacji).

`pinMode(nr wyjścia, tryb pracy);`

nr wyjścia - to liczba od 0 do 53 - przy czym na tablicy korzystamy z wyjść od 2 do 13

tryb pracy:

INPUT - piny jako wejścia

OUTPUT - piny jako wyjścia

INPUT_PULLUP - piny jako wejścia z nastawionym „z góry” stanem wysokim (czyli na wejściu jest napięcie 5V).

W naszej przykładowej lekcji ustawiamy pin nr 2 jako wyjście - czyli mikrokontroler załącza albo wyłącza napięcie na tym pinie. Dzięki temu możemy sterować pracą różnych elementów podłączonych do tego pinu.

6.3 Instrukcja `digitalWrite()`:

Najważniejsza instrukcja sterująca stanem pinów mikrokontrolera, gdyż za jej pomocą możemy włączyć napięcie albo je wyłączyć na danym pinie.

`digitalWrite(2,1);` oznacza, iż pin nr 2 zostanie ustawiony w stan wysoki (napięcie na wyjściu 5V)

`digitalWrite(2,0);` oznacza, iż pin nr 2 zostanie ustawiony w stan niski (napięcie na wyjściu 0V)

Analogicznie każdy inny pin możesz przy pomocy tej instrukcji ustawić jako wysoki albo niski.

Łącząc pin mikrokontrolera z dowolną diodą uzyskać efekt jej załączenia albo wyłączenia.

6.4 Instrukcja `delay()`:

Ta instrukcja zatrzymuje działanie mikrokontrolera na określony czas (w milisekundach).

1000 milisekund = 1 sekunda

Zapis: `delay(2000);` oznacza, że mikrokontroler zatrzyma działanie programu na 2 sekundy.

Efekt działania programu:

w funkcji `setup`:

`pinMode(2, OUTPUT);` - wybraliśmy pin 2 i ustawiamy go jako wyjście

w funkcji `loop`:

`digitalWrite(2,1);` - pin nr 2 przyjmuje wartość wysoką (jest napięcie - zapala się dioda podłączona do tego pinu)

`delay(500);` - pauza trwająca pół sekundy - cały ten czas stan pinu nr 2 nie zmienia się, więc dioda się świeci.

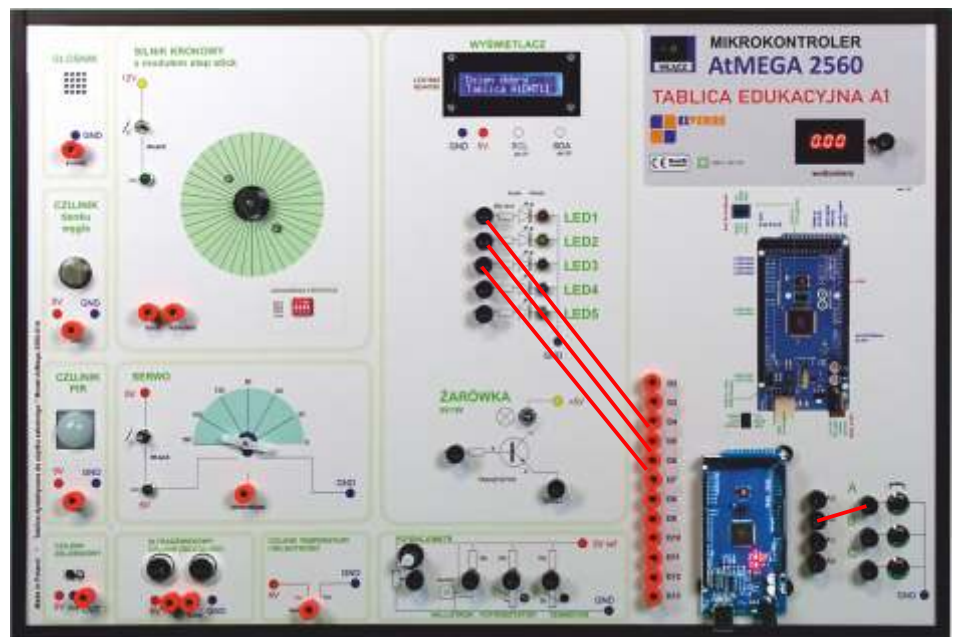
`digitalWrite(2,0);` - zmieniamy stan pinu nr 2 na niski, napięcie na pinie wynosi 0V, więc dioda gaśnie

`delay(500);` - pauza trwająca pół sekundy

6.4.1 Zadanie / `delay()`:

Zmieniaj czas trwania świecenia albo czas trwania wygaszenia diody zmieniając wartość opóźnienia tj. liczbę podaną w nawiasie polecenia **`delay`**

6. Lekcje programowania



6.5 LEKCJA 2: Pętla while, oczekiwanie na wciśnięcie przycisku. Sygnalizacja świetlna

```
#define RED 4
#define GREEN 5
#define ORANGE 10
#define BUTTON 2
void setup() {
    // moj program
    pinMode(RED,OUTPUT);
    pinMode (ORANGE,OUTPUT);
    pinMode(GREEN, OUTPUT);
    pinMode(BUTTON, INPUT_PULLUP);
}

void loop() {
    // put your main code here, to run repeatedly:
    digitalWrite(RED,1);
    while(digitalRead(BUTTON)==1){
        delay(1);
    }

    digitalWrite (ORANGE,1);
    delay(2000);
    digitalWrite(RED,0);
    digitalWrite(ORANGE,0);
    digitalWrite(GREEN,1);

    for (int czas=0; czas<25; czas++){
        tone(8,800,120);
        delay(200);
    }

    digitalWrite(GREEN,0);
    digitalWrite(ORANGE,1);
    delay(1000);
    digitalWrite(ORANGE,0);
}
```

6. Lekcje programowania

6.6 Instrukcja #define

#define zmienna wartość instrukcja pozwala zdefiniować stałą wartość dla używanej w programie zmiennej (wartości ukrytej).

przykład `#define RED 4` - dla zmiennej `RED` ustaliliśmy wartość `4` - nie może ona być zmieniona w programie

Dzięki definicjom upraszczamy strukturę programu, czyniąc go bardziej przejrzystą i niezawodną.

6.7 Instrukcja `digitalRead(nr pinu);`

Ta instrukcja pozwala odczytać wartość dowolnego (cyfrowego) pinu mikrokontrolera. Wartość ta może być przypisana do zmiennej albo porównywana instrukcjami warunkowymi.

```
int bool=digitalRead(2);
```

Do zmiennej (która może być tylko rodzaju **bool** - *boolean* czyli liczbą zero albo jeden) przypisujemy wartość odczytaną z pinu nr 2 mikrokontrolera.

6.8 Pętla warunkowa `while();`

Instrukcja `while()` należy do tzw. pętli warunkowych.

Polecenia zawarte wewnątrz pętli - nawiasy `{ }` będą wykonywane dopóki spełniony jest warunek pętli

przykład

```
while (digitalRead(2)==1){  
// dopóki pin nr 2 ma stan wysoki wykonuj to co jest tutaj  
}
```

W ten sposób możemy zatrzymać działanie programu tak długo, dopóki stan przycisku podłączonego do pinu nr 2 nie zmieni się (na zero). Przyciśnięcie przycisku spowoduje połączenie pinu z masą układu, czyli podanie logicznego zera.

6.9 Pętla warunkowa `for();`

Pętla będzie wykonywana dopóki zmienna użyta jako argument pętli, nie osiągnie zadanej wartości.

```
for (int czas=0; czas<25; czas++){  
tone(8,800,120);  
delay(200);  
}
```

Zmienne `czas` startuje od wartości 0. Pętla będzie wykonywana dopóki zmienna `czas` ma wartość mniejszą niż 25. Zapis `czas++` - oznacza, iż każdy powrót na początek pętli oznacza zwiększenie wartości zmiennej `czas` o jeden.

6. Lekcje programowania

W niniejszej instrukcji opisaliśmy podstawowe zasady programowania mikrokontrolerów w środowisku ARDUINO. Zachęcamy do pogłębiania wiedzy z zakresu obsługi i programowania mikrokontrolerów dzięki dostępnym darmowym, przykładowym lekcjom z użyciem tej tablicy edukacyjnej, a także do podniesienia swoich kwalifikacji w zakresie języka programowania mikrokontrolerów C oraz SCRATCH na naszych kursach dla nauczycieli.

Więcej informacji zasięgniesz u swoich dostawców oraz na stronie laboratoria-przyszlosci.edu.pl