SCENARIUSZ LEKCJI

„Programowanie płytki Arduino. Symulacja alarmu”

Scenariusz opracowany w ramach projektu
„Powiślańska Szkoła Ćwiczeń – Kwidzyn”,
współfinansowanego ze środków Europejskiego Funduszu Społecznego
w ramach Programu Operacyjnego Wiedza Edukacja Rozwój 2014-2020.

SCENARIUSZ LEKCJI

1. **Przedmiot**: informatyka
2. **Etap edukacyjny**: III
3. **Klasa:** 2
4. **Czas trwania**: 45 minut
5. **Temat zajęć/lekcji**: Programowanie płytki Arduino. Symulacja alarmu.
6. **Cele zajęć/lekcji:**

Cel ogólny: Stworzenie programu symulującego, aktywację alarmu gdy otwarte okno.

Cele operacyjne:

Uczeń potrafi:

- wykorzystać aplikację Arduino do programowania w języku C,
- opisuje różnice pomiędzy programistycznym środowiskiem lokalnym a webowym,
- dostrzega różnice pomiędzy tradycyjnym kompilatorem a Arduino
- utworzyć program z wykorzystaniem:
    - instrukcji warunkowej
    - Instrukcji pętli
    - Definiowania funkcji
    - Korzystania z generatorów liczb
- przeprowadzi modyfikację programu,
- przeprowadzić test napisanego programu i wyeliminować błędy,
- pobierze wyniki swojej pracy na dysk komputera i udostępni je w sieci.

7. **Treści nauczania z podstawy programowej realizowane w czasie zajęć/lekcji:**

Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:

- planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komutacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie

rozwiązania, zaprogramowanie i testowanie rozwiązania) (I.1),

- porównuje działanie różnych algorytmów dla wybranego problemu, analizuje algorytmy na podstawie ich gotowych implementacji (I.4).

Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:

- do realizacji rozwiązań problemów prawidłowo dobiera środowiska informatyczne, aplikacje oraz zasoby, wykorzystuje również elementy robotyki (II.2)

Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi. Uczeń:

- objaśnia funkcje innych niż komputer urządzeń cyfrowych i korzysta z ich możliwości(III.2)

Rozwijanie kompetencji społecznych. Uczeń:

- poszerza i uzupełnia swoją wiedzę korzystając z zasobów udostępnionych na platformach do e-nauczania. (IV.6)

8. Metody pracy z uczniami (z uwzględnieniem uczniów o specjalnych potrzebach edukacyjnych):

- pokaz z objaśnieniem,, ćwiczenia, praca w grupie, gamyfikacja.

9. Środki dydaktyczne wykorzystane przez nauczyciela i przez uczniów:

- laptop bądź iPad z dostępem do Internetu ,
- środowisko Arduino,
- płytka Arduino z zestawem do automatyki(płytka stykowa, przewody, włącznik, diody)
- projektor z ekranem lub tablica interaktywna,

Przebieg lekcji:

Część wprowadzająca: (około 10 min)

Nauczyciel przedstawia temat zajęć, omawia zasadę działania środowiska Arduino oraz płytki .

Część właściwa: (około 30 min)

1. Uczniowie uruchamiają środowisko Arduino i nawiązują połączenie z płytką.

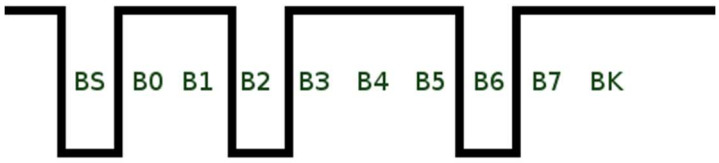
2. Nauczyciel omawia sposób komunikacji.

3. UART (komunikacja z PC).

Omawiane Arduino pozwala na wykorzystanie licznych interfejsów komunikacyjnych. W tej części zajmiemy się UARTem. Jest to prosty i bardzo popularny interfejs szeregowy.

a. Jak działa UART?

Jego zasada działania opiera się na szeregowym wysłaniu ciągu bitów, które następnie składane są w informację. Pojedyncza ramka (w uproszczeniu bajt) danych jest nadawany w poniższej postaci:



Informacja na temat systemów numerycznych w szczególności – system binarny.

Transmisja rozpoczyna się od bitu startu, zaznaczonego na rysunku jako BS. Zawsze jest to bit będący logicznym zerem. Następnie, zależnie od konfiguracji, następuje po sobie 7, 8 lub 9 bitów danych (tutaj zaznaczone jako B0B7), które są wysyłaną informacją. Bit stopu (zaznaczony tutaj jako bit BK) to bit będący logiczną jedynką – mówi o końcu transmisji.

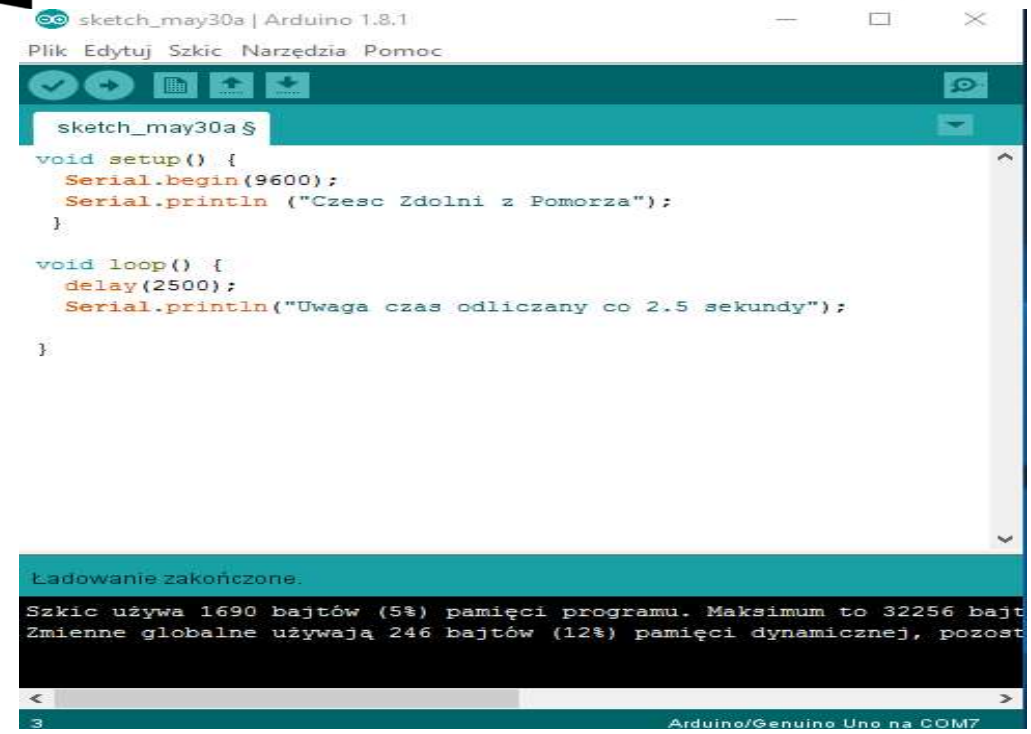
Wykorzystując UART w Arduino interesują nas dwa piny:

Tx do wysyłania danych (pin 1 w Arduino),
Rx do odbierania danych (pin 0 w Arduino).

Komputer, z którym chcemy nawiązać łączność, musi być również wyposażony w odpowiedni interfejs USB, które znacznie ułatwiają sprawę. Korzystając z Arduino nie trzeba się o to martwić. Konwerter taki został już wbudowany w nasza płytkę. Przejdźmy do przykładów praktycznych. Pierwsze dwa programy wymagają TYLKO podłączenia Arduino przez USB z komputerem. Dopiero później dodamy do naszego układu peryferia.

Tekst z układu do komputera.

Zadaniem poniższego programu jest proste, regularne wysyłanie tekstu do komputera:

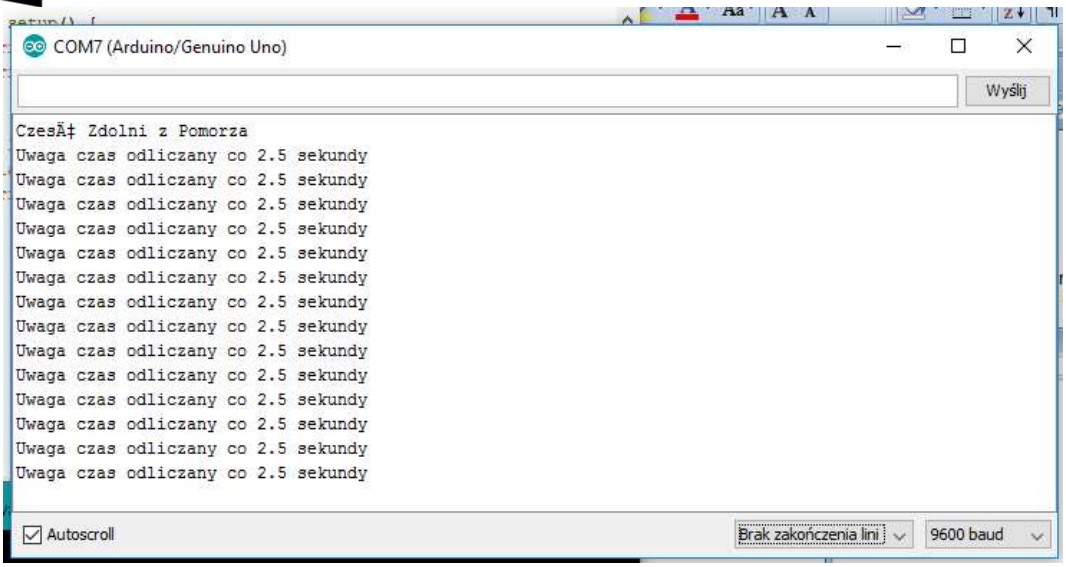


Po wgraniu powyższego programu pozornie nic nie będzie się działo. Aby zaobserwować jego działanie, musimy z menu Arduino wybrać:

Narzędzia> *Monitor Portu Szeregowego*.

Dzięki temu otworzy się nowe okno. Będzie to potocznie zwany terminal. W tym miejscu możemy obserwować, to co jest przesyłane do/z Arduino przez port COM, czyli nasz UART.

Ważne, aby terminal pracował z odpowiednią prędkością na porcie COM, do którego podłączone jest Arduino! Ustawienia prędkości znaleźć można w prawym, dolnym rogu terminala.



Przejdźmy do analizy programu. Pierwszą rzeczą, którą należy wykonać jest ustawienie prędkości transmisji. Służy do tego funkcja:

Serial.begin(speed), gdzie **speed** określa prędkość transmisji. Ustawmy ją na **9600 baud/sec**.

Natomiast do wysyłania ciągu znaków służy funkcja:

Serial.println(val), gdzie **val** to ciąg znaków lub liczba.

Tekst „Witaj ….” zostaje wyświetlony tylko raz, ponieważ został umieszczony w funkcji setup, instrukcje tam zawarte wykonywane są tylko po uruchomieniu programu.

Działającą transmisję możemy również zaobserwować na diodach wbudowanych w Arduino (oznaczone jako Tx oraz Rx)! Świecą one, gdy trwa przesyłanie danych do/z naszej płytki.

Zadanie 1

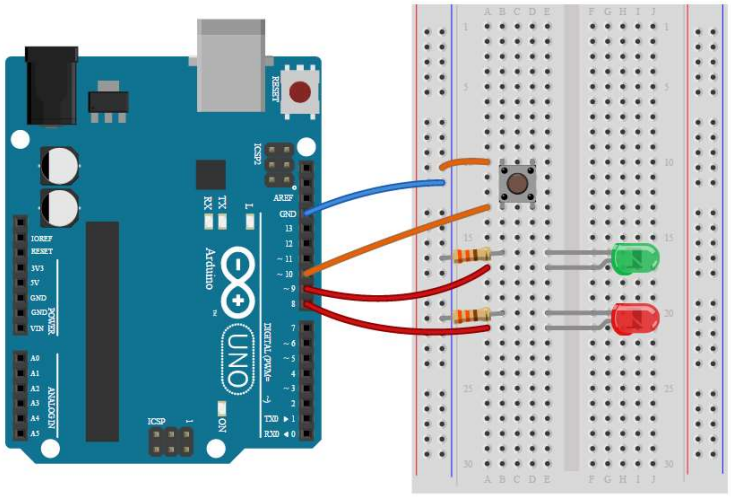
Sprawdź co stanie się, gdy w oknie terminala będziesz wybierał prędkości inne od ustawionych w Arduino. Kiedy pojawiają się błędy? W jaki sposób się objawiają? Jak sam zobaczysz błędy są dość charakterystyczne, warto o tym pamiętać.

a. Interakcja z programem

Oczywiście informacje nie muszą być wysyłane przez UART cały czas, nadawanie i odbieranie może również przebiegać jednorazowo w wybranej przez nas chwili. Jest to bardzo użyteczne, chociażby do diagnozowania pracy układu lub sygnalizacji różnych zdarzeń. Wyobraźmy sobie

sytuację, w której Arduino byłoby zamontowane w roli sygnalizatora otwarcia okna, do którego ramy zamontowano specjalny przycisk/czujnik. Cały mechanizm działa w taki sposób, że gdy okno jest zamknięte przycisk zwierany jest do masy, w przeciwnym wypadku obwód jest przerwany. Korzystając z poprzednio zdobytej wiedzy mógłbyś napisać program, który włącza kolorową diodę, gdy okno jest otwarte. Spróbujmy jednak rozbudować nasz program. Wyposażmy Arduino w przycisk (imitujący sensor w ramie okna) oraz dwie diody (zieloną i czerwoną).

Budowa układu



Gdy okno jest zamknięte (przycisk wciśnięty) świeci się zielona dioda. Gdy przerwiemy obwód (przestaniemy wciskać przycisk) powinna włączyć się czerwona dioda, a w terminalu powinniśmy odczytać komunikat *„Uwaga! Alarm! Okno nie jest zamknięte!”*.

Arduino

```
1  void setup(){
2    Serial.begin(9600); //Uruchamiamy transmisję
3
4    pinMode(8, OUTPUT); //Wyjście diody czerwonej
5    pinMode(9, OUTPUT); //Wyjście diody zielonej
6    pinMode(10, INPUT_PULLUP); //Przycisk
7
8    digitalWrite(8, LOW); //Wyłączenie obu diod
9    digitalWrite(9, LOW);
10 }
11
12 void loop() {
13   if (digitalRead(10) == LOW) { //Jeśli przycisk jest wciśnięty
14     digitalWrite(9, HIGH); //Włączenie diody zielonej
15     digitalWrite(8, LOW); //Wyłączenie diody czerwonej
16   } else { //Jeśli przycisk nie jest wciśnięty
17     digitalWrite(9, LOW); //Wyłączenie diody zielonej
18     digitalWrite(8, HIGH); //Włączenie diody czerwonej
19     Serial.println("Uwaga! Alarm! Okno nie jest zamkniete!");
20   }
21 }
```

Zadanie 2.

Gdy okno nie jest zamknięte informacja o alarmie nadawana jest cały czas. Wolelibyśmy jednak, aby wysyłana była tylko raz . Rozwiąż ten problem korzystając z pętli **while**.

Podsumowanie i ewaluacja (5min.)

Nauczyciel zadaje uczniom pytania:

• Co najbardziej podobało się Wam podczas dzisiejszej lekcji?

• Z czym mieliście największe problemy?

• Do czego można wykorzystać umiejętności zdobyte na tej lekcji?

Bibliografia:

1. Podstawa programowa do szkoły ponadpodstawowowej do przedmiotu informatyka,
2. https://forbot.pl
3. https://eduinf.waw.pl/