

MoboLab – roboty i tablety w Twojej szkole
Obszar II. „Stwórz własnego robota”
Scenariusze lekcji i zajęć pozalekcyjnych

SCENARIUSZ 13. GRA „REWOLWEROWIEC”

scenariusz zajęć pozalekcyjnych

autor: Wojciech Karcz

redakcja: Agnieszka Koszowska

SŁOWA KLUCZOWE:

Arduino, gra, automat, przycisk, dioda LED

KRÓTKI OPIS ZAJĘĆ:

Arduino może też zostać wykorzystane jako urządzenie do budowania prostych automatów do gier. Podczas zajęć uczniowie i uczennice budują grę, w której wygrywa ten, kto pierwszy zareaguje – poprzez wciśnięcie przycisku – na zapalającą się (w losowym momencie) diodę.

WIEDZA I UMIEJĘTNOŚCI ZDOBYTE PRZEZ UCZNIĄ / UCZENNICĘ:

- wie, czym są mikrokontrolery i do czego służą,
- zna pojęcia: mikrokontroler, skrypt, program, algorytm, sterowanie, warunek, pętla,
- zna projekt Arduino, wie, czym jest platforma Arduino, z jakich części się składa,
- potrafi w podstawowym stopniu samodzielnie obsługiwać Arduino (podłączyć płytkę do komputera, wgrać prosty program),
- wie, co to jest dioda LED,
- potrafi poprawnie podłączyć diodę LED do Arduino,
- zna podstawowe elementy interfejsu środowiska programistycznego Arduino IDE i podstawowe komendy języka Arduino IDE: `pinMode()`, `digitalWrite()`, `delay()`,
- rozumie zasadę działania funkcji `digitalWrite()` i potrafi wykorzystać ją w praktyce,
- wie, jak łączyć przyciski oraz diody LED w celu programowania prostej gry,

- potrafi stworzyć prosty plan programu, który będzie obsługiwał grę,
- zna podstawowe elementy języka **Scratch**, potrafi stworzyć prosty skrypt w tym języku.

GRUPA DOCELOWA:

Starsze klasy szkoły podstawowej (VII-) i klasy gimnazjalne (po dostosowaniu: możliwość realizacji w młodszych klasach: I-III i IV-VI szkoły podstawowej). W młodszych klasach – możliwość wykorzystania programu mBlock (po przejściu scenariusza nr 18. *Programowanie Arduino z wykorzystaniem programu mBlock*) lub Scratch for Arduino (po przejściu scenariusza nr 1. *Wprowadzenie do Arduino*).

LICZBA UCZNIÓW/UCZENNIC W GRUPIE:

Liczba optymalna: 12, liczba maksymalna: 16

CZAS TRWANIA ZAJĘĆ:

90 min (lub 2 x 45 minut)

STOPIEŃ TRUDNOŚCI/SKOMPLIKOWANIA

(w skali od 1 do 5 dla obszaru II. „Stwórz własnego robota”):

4

POTRZEBNY SPRZĘT I OPROGRAMOWANIE:

- komputer (przenośny lub stacjonarny),
- program Arduino IDE (do pobrania ze strony: <http://www.arduino.org/downloads>),
- (opcjonalnie) program mBlock (do pobrania ze strony: <http://www.mblock.cc/download/>) lub Scratch for Arduino (do pobrania ze strony: <http://s4a.cat/>),
- płytki Arduino UNO i kabel USB A-B (dla każdego uczestnika lub dla pary uczestników),
- płytki stykowe,
- oporniki 220 omów oraz 10K omów,
- przewody połączeniowe,
- przyciski („micro switch” THT),
- diody LED w różnych kolorach,
- projektor i laptop (w części teoretycznej).

CO NALEŻY PRZYGOTOWAĆ PRZED ZAJĘCIAMI:

- zainstalować program Arduino IDE,
- (opcjonalnie): zainstalować program **mBlock** lub **Scratch for Arduino**,
- sprawdzić, czy wszystkie komputery wykrywają podłączone Arduino,
- przeczytać dokładnie scenariusz,
- zapoznać się z materiałami dodatkowymi (w części „Pigułka wiedzy i inspiracji”),
- wykonać samodzielnie zadania zawarte w scenariuszu,
- przy każdym stanowisku komputerowym rozłożyć elementy zestawu Arduino, które będą wykorzystywane na tych zajęciach,
- dopasować stopień trudności zadania do potrzeb i możliwości klasy, dla której organizowane są zajęcia według wskazówek zawartych w scenariuszu.

KOMPETENCJE OSOBY PROWADZĄCEJ:

- wie, czym jest projekt Arduino, zna podstawowe informacje o projekcie,
- potrafi przynajmniej w stopniu podstawowym obsługiwać Arduino,
- zna podstawowe pojęcia z zakresu elektroniki,
- zna podstawowe pojęcia programistyczne,
- wie, dlaczego warto uczyć się programowania i jakie korzyści daje posiadanie umiejętności programistycznych,
- potrafi zachęcić do nauki programowania zarówno chłopców, jak i dziewczynki.

PRZEBIEG ZAJĘĆ:

Podłączenie Arduino, uruchomienie programu Arduino IDE i przypomnienie podstawowych informacji – ok. 15 minut

Uwaga! Informacje o tym, jak podłączyć Arduino, uruchomić program Arduino IDE i Scratch for Arduino, a także podstawowe informacje niezbędne przy rozpoczynaniu pracy z Arduino zawierają scenariusze 1 i 2. Tę część zajęć warto powtarzać za każdym razem w takim zakresie, jaki jest potrzebny, do czasu aż podstawowy materiał zostanie utrwalony.

Wstęp, czyli jak za pomocą Arduino stworzyć grę – 15 min

Te zajęcia są poświęcone w całości zaprojektowaniu i wykonaniu prostej gry na Arduino. Arduino służy nie tylko do budowania różnych urządzeń wykonujących ściśle określone użyteczne zadania. Mównieź też wykorzystać je do konstruowania automatów do gier.

Rozpoczynamy od dyskusji z uczniami na temat gier. Zadajemy takie pytania, jak:

- *Jak działa gra komputerowa?*
- *Czego potrzebujemy, by zagrać w grę?* (chodzi przede wszystkim o kontrolery, takie jak klawiatura, pad, joystick)
- *Co to są automaty do gier?*
- *Jak powstają gry?*

Podsumowujemy dyskusję i przedstawiamy temat dzisiejszych zajęć, którym jest budowa gry na Arduino. Gra nazywa się „Rewolwerowiec” i polega na tym, że wygrywa ta osoba, która pierwsza wciśnie swój przycisk po zapaleniu się czerwonej diody LED. Nazwa nawiązuje do Dzikiego Zachodu i pojedynków na rewolwery, gdzie kluczową rolę odgrywał refleks i czas reakcji kowboja.

Gra, którą chcemy zbudować ma działać według poniższego schematu:

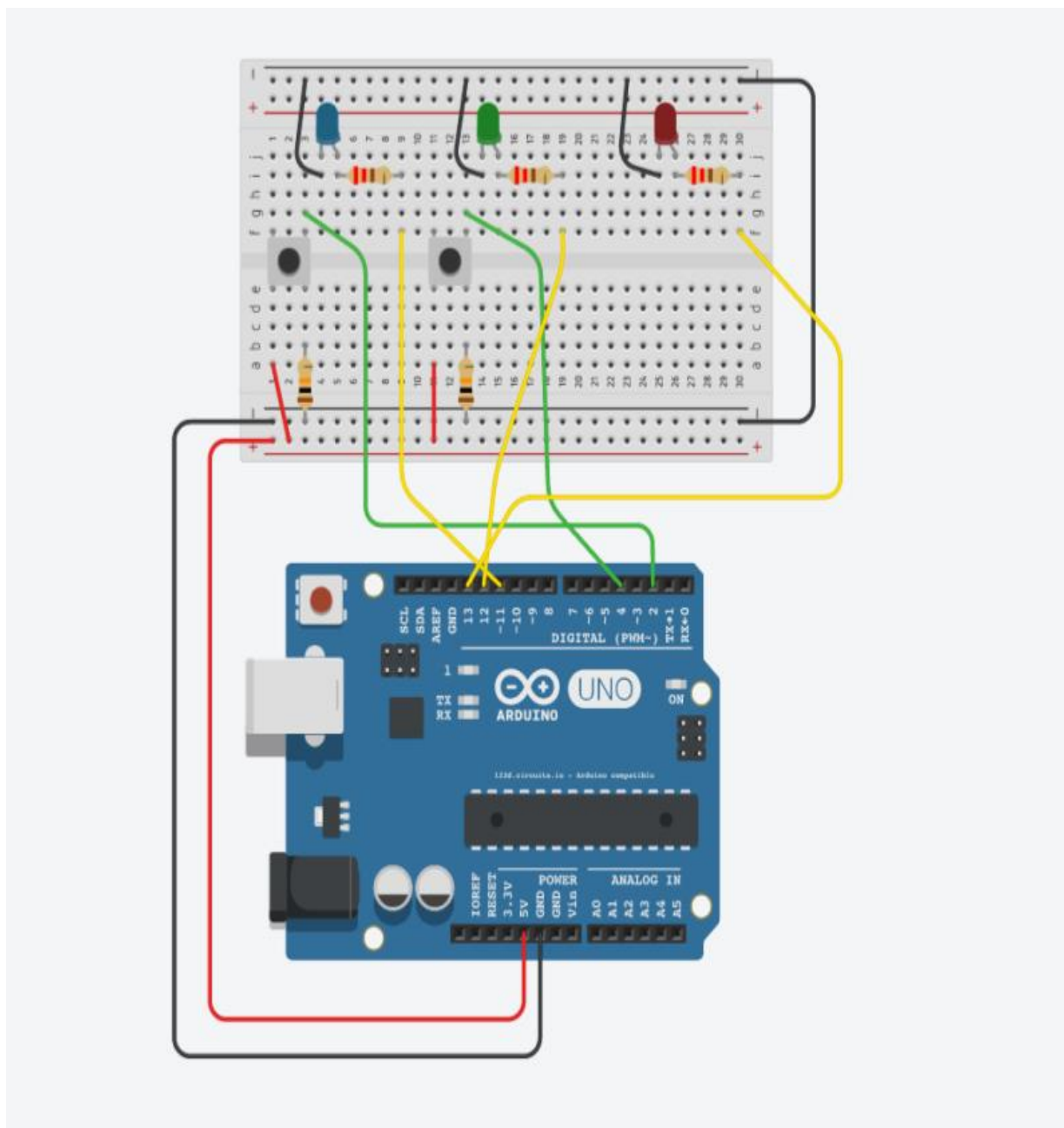
- naciskamy przycisk startu,
- otrzymujemy komunikat o rozpoczęciu odliczania,
- rozpoczyna się odliczanie do zapalenia diody (czas jest za każdym razem ustalany losowo),
- zapala się dioda LED,
- gracze wciskają (każdy gracz wciska swój przycisk),
- program wykrywa, który gracz był pierwszy i zapala się dioda LED obok jego przycisku, dodatkowo pojawia się komunikat, który gracz wygrał,
- naciskamy przycisk RESET na Arduino i rozpoczynamy nową grę.

Tłumaczymy uczniom, że przy tworzeniu takich zaawansowanych projektów na Arduino, trzeba rozbić je na mniejsze elementy, które są łatwiejsze w realizacji, a dopiero potem składamy wszystko w jeden program. W ten sposób jest realizowana większość projektów programistycznych. Rzadko od razu rozpoczynamy pracę nad

całym programem. W pierwszej kolejności zajmujemy się pojedynczymi klockami, z których powstanie cała konstrukcja programu.

Montaż układu – 15 min

Poniżej znajduje się schemat układu, który montujemy wspólnie z uczniami:



W przeciwieństwie do wielu innych projektów, w tym przypadku skupimy się przede wszystkim na warstwie programistycznej.

Warto w tym miejscu przywołać podstawowe informacje o przyciskach, kontrolowaniu wyjść cyfrowych, czy odczytywaniu wartości na pinach wejściowych.

W tym miejscu możliwy jest podział zajęć na dwie części (kolejna część scenariusza będzie realizowana na następnych zajęciach).

Przypomnienie materiału, odtworzenie układu – 10 minut

Rozpoczynamy od krótkiego przypomnienia materiału z poprzedniej części zajęć i odtworzenia układu zbudowanego na poprzednich zajęciach.

Programowanie gry – 35 minut

Na następnych stronach znajduje się pełen kod programu gry. W dalszej części scenariusza znajduje się dokładny opis każdej sekcji odpowiedzialnej za różne zadania.

```
int APlayerInput = 2;
int BPlayerInput = 4;

int APlayerLed = 11;
int BPlayerLed = 12;

int RandomLed = 13;

int APlayerState = LOW;
int BPlayerState = LOW;

unsigned long randNumber;
unsigned long minRandomNumber = 2000;
unsigned long maxRandomNumber = 6000;

void setup() {
  pinMode(APlayerInput, INPUT);
  pinMode(BPlayerInput, INPUT);

  pinMode(APlayerLed, OUTPUT);
  pinMode(BPlayerLed, OUTPUT);
  pinMode(RandomLed, OUTPUT);

  Serial.begin(115200);

  digitalWrite(APlayerLed, LOW);
  digitalWrite(BPlayerLed, LOW);
  digitalWrite(RandomLed, LOW);
}
```

```

    randomSeed(analogRead(0));
    randomNumber = random(minRandomNumber,maxRandomNumber);

    Serial.println("Przygotuj sie do gry...");
    delay(2000);
    Serial.println("START!");
    delay(randNumber);
    digitalWrite(RandomLed, HIGH);

}

void loop() {
    APlayerState = digitalRead(APlayerInput);
    BPlayerState = digitalRead(BPlayerInput);

    if (APlayerState == HIGH) {
        digitalWrite(APlayerLed, HIGH);
        Serial.println("Wygral gracz A");
        delay(20000);
    }
    else if (BPlayerState == HIGH) {
        digitalWrite(BPlayerLed, HIGH);
        Serial.println("Wygral gracz B");
        delay(20000);
    }

    delay(20);
}

```

Program piszemy wspólnie z uczniami na bieżąco tłumacząc, jak działa każda sekcja. Najlepiej po każdej części od razu sprawdzać program wgrywając go na Arduino. Warto również kompilować co jakiś czas nasz program w poszukiwaniu ewentualnych błędów.

Definiowanie zmiennych

definiujemy piny wejściowe
(przyciski graczy)

```
int APlayerInput = 2;  
int BPlayerInput = 4;
```

definiujemy piny wyjściowe
(diody LED)

```
int APlayerLed = 11;  
int BPlayerLed = 12;
```

definiujemy zmienne
przechowujące informacje
o wciśniętych przyciskach

```
int RandomLed = 13;
```

```
int APlayerState = LOW;  
int BPlayerState = LOW;
```

definiujemy zmienne
przechowujące losowy czas
odliczania

```
unsigned long randomNumber;  
unsigned long minRandomNumber = 2000;  
unsigned long maxRandomNumber = 6000;
```

definiujemy zakres liczby
losowej

Źródło: Materiały własne

Podobnie jak w wielu innych programach, na samym początku definiujemy zmienne. W tym wypadku jest ich dosyć dużo, program jest dosyć rozbudowany i potrzebujemy sporo różnych parametrów. Nasze zmienne to:

APlayerInput, BPlayerInput – zmienne oznaczające piny wejściowe dla przycisków przypisanych odpowiednio do gracza A oraz B.

APlayerLed, BPlayerLed – zmienne określające numery pinów, do których podłączone są odpowiadające graczom diody LED.

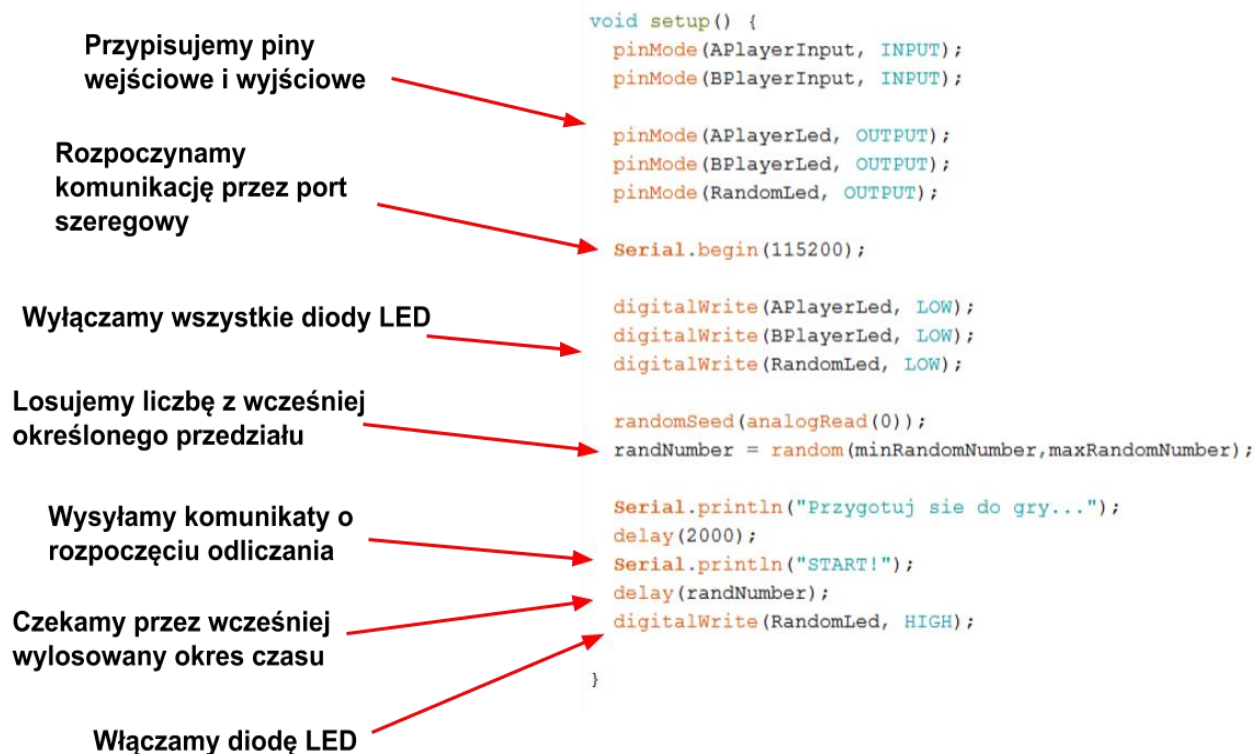
RandomLed – pin, do którego podłączamy diodę oznaczającą rozpoczęcie pojedynku (od momentu zapalenia się tej diody gracze wciskają przyciski).

APlayerState, BPlayerState – zmienne przechowujące stan na pinach wejściowych dla przycisków graczy A i B (tutaj przechowujemy informacje o tym, czy przycisk jest wciśnięty, czy nie).

randNumber – losowa liczba określająca czas od startu do zapalenia diody.

minRandomNumber, maxRandomNumber – tutaj określamy zakres, w jakim ma się losować liczba randNumber.

Sekwencja startowa

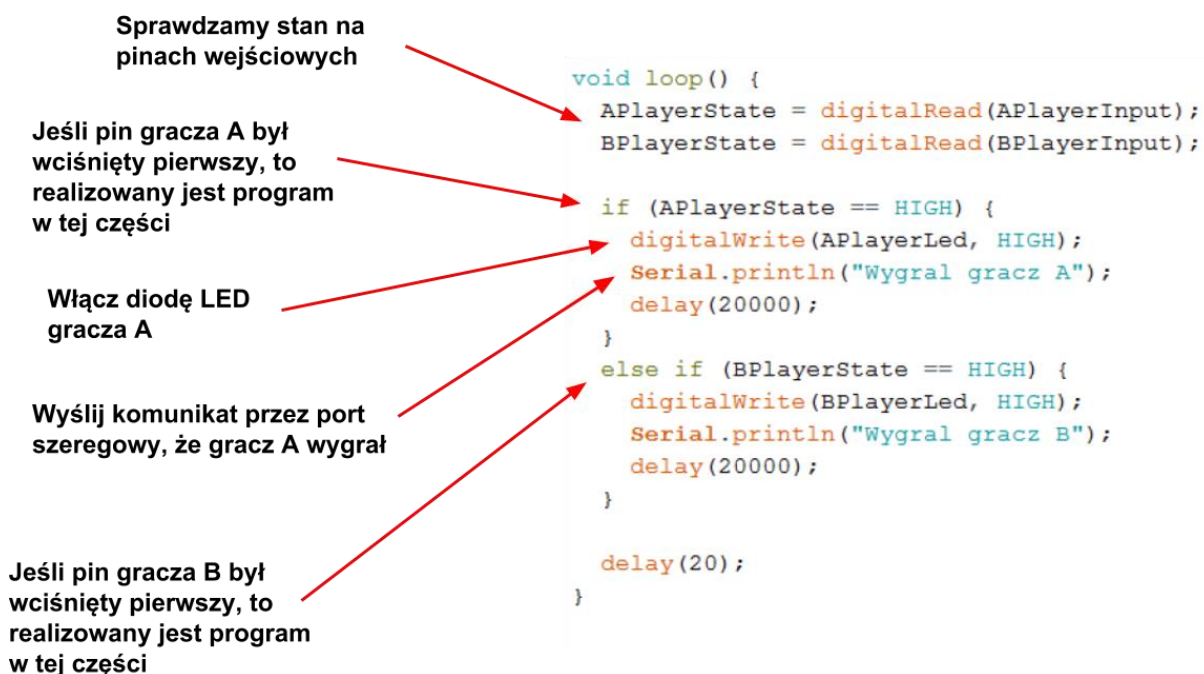


Po przypisaniu zmiennych przechodzimy do procedury startowej, która jest wykonywana tylko jeden raz przy każdorazowym uruchomieniu Arduino lub wciśnięciu przycisku RESET na Arduino. Aby uprościć maksymalnie działanie programu, wykorzystamy wbudowany przycisk RESET w Arduino. Po każdym przyciśnięciu przycisku, nasz program będzie uruchamiany od nowa. Wszystkie funkcje znajdujące się w części void setup() zostaną wykonane tylko raz. Potem w pętli void loop() sprawdzamy, czy któryś z graczy nie wcisnął swojego przycisku. Kiedy chcemy rozpocząć grę od nowa, to wciskamy przycisk RESET.

Na początku sekcji void setup() definiujemy piny wejściowe (dla przycisków) oraz wyjściowe (dla diod). Rozpoczynamy komunikację przez port szeregowy oraz wyłączamy dla pewności wszystkie diody LED.

Następnie uruchamiamy funkcję randomSeed(), która inicjalizuje generator liczb pseudolosowych. W kolejnym kroku przypisujemy do zmiennej randomNumber losową liczbę z wcześniej zdefiniowanego zakresu. Więcej na temat zasady działania generatora liczb pseudolosowych w Arduino znajduje się w materiałach dodatkowych. Ostatnie funkcje w tej części to już ostatnia sekwencja startowa. Wysyłamy przez port szeregowy informacje „Przygotuj się do gry...”, czekamy 2s, „START”. Od tego momentu rusza nasze losowe generowane odliczanie za pomocą funkcji delay(), gdzie argumentem jest zmienna randomNumber. Po skończonym odliczaniu zapala się dioda LED. W tym momencie gracze próbują jak najszybciej wcisnąć przycisk.

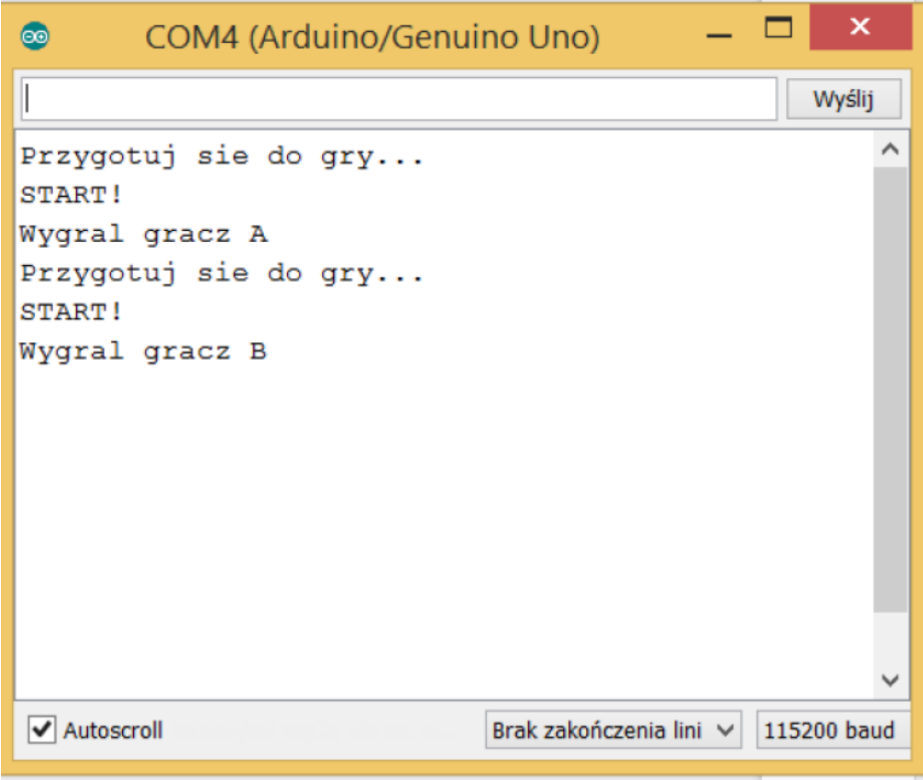
Sprawdzanie, kto wygrał



W ostatniej części naszego programu naszym zadaniem jest sprawdzenie, kto pierwszy wcisnął przycisk oraz przekazanie tej informacji graczom w jakiś widoczny sposób.

Główną pętlę naszego programu rozpoczynamy od sprawdzenia stanu pinów wejściowych połączonych z dwoma przyciskami poprzez funkcję `digitalRead()`. Następnie przechodzimy do dwóch kolejno po sobie następujących instrukcji warunkowych. W pierwszej sprawdzamy, że jeśli gracz A wcisnął pierwszy przycisk (wykryliśmy stan wysoki na pinie `APlayerInput`), zapala się dioda LED przypisana do gracza A (`APlayerLed`) oraz wysyła się informacja przez port szeregowy, że gracz A wygrał. W drugiej instrukcji warunkowej zaczynającej się od „if else” zasada działania jest analogiczna, tylko dotyczy gracza B.

Kiedy chcemy zagrać ponownie, to musimy wcisnąć przycisk RESET na Arduino i program uruchomi się od początku.



The screenshot shows the serial monitor window for COM4 (Arduino/Genuino Uno). The window title is "COM4 (Arduino/Genuino Uno)". The main area displays the following text: "Przygotuj sie do gry...", "START!", "Wygral gracz A", "Przygotuj sie do gry...", "START!", and "Wygral gracz B". The text is displayed in a monospaced font. At the top right of the window is a "Wyślij" button. At the bottom of the window, there are three controls: a checked "Autoscroll" checkbox, a dropdown menu set to "Brak zakończenia lini", and a baud rate selector set to "115200 baud".

Po napisaniu programu sprawdzamy z uczniami, czy wszystko działa. Przy takich już bardziej zaawansowanych projektach bardzo często mogą pojawić się różne problemy, błędy. To jest normalna sytuacja i nie należy się nią przejmować, tylko trzeba wielokrotnie oraz bardzo dokładnie analizować kod w poszukiwaniu błędów. Poniżej znajdują się najczęstsze problemy, które mogą wystąpić przy konstruowaniu takiej gry, i wskazówki, jak im przeciwdziałać:

1. Dokładnie przeglądajmy kod programu i sprawdzajmy komunikaty, które

pojawiają się przy kompilowaniu na dole Arduino IDE.

2. Sprawdźmy, czy wszystkie zmienne są przypisane do pinów, gdzie faktycznie są podłączone przewody.
3. Sprawdźmy bardzo dokładnie wszystkie połączenia na płytce stykowej oraz w Arduino. Bardzo często nawet niewielka pomyłka unieruchamia nasz program.
4. W newralgicznych punktach programu warto do debugowania (więcej informacji w materiałach dodatkowych) wykorzystać polecenie `Serial.println()`. Na przykład chcemy sprawdzić, czy losowa liczba generuje nam się prawidłowo, to zaraz po jej wylosowaniu wysyłamy ją przez port szeregowy i podglądamy w Szeregowym monitorze, czy wszystko działa prawidłowo.

MOŻLIWE MODYFIKACJE DLA MŁODSZYCH KLAS:

Pracując z uczniami w młodszych klasach można wykorzystać zamiast Arduino IDE program S4A (Arduino for Scratch). W przypadku zajęć z młodszymi dziećmi warto zwrócić uwagę na ewentualne problemy z dokładnym podłączaniem przewodów.

ZADANIE SPRAWDZAJĄCE UMIEJĘTNOŚCI ZDOBYTE PODCZAS ZAJĘĆ:

Uczeń / uczennica, pracując samodzielnie albo w dwu- lub trzyosobowym zespole tworzy grę, w której wygrywa ten, kto pierwszy zareaguje – poprzez wciśnięcie przycisku – na zapalającą się (w losowym momencie) diodę. Dla bardziej zaawansowanych uczniów można przygotować zadanie dodatkowe, polegające na zmianie reakcji diody LED na wciśnięcie przycisku przez jednego z graczy (zamiast ciągle świecenia dioda ma migać przez pewien czas).

PIGUŁKA WIEDZY I INSPIRACJI DLA OSÓB PROWADZĄCYCH:

Kurs programowania Arduino Forbot:

<http://forbot.pl/blog/artykuly/programowanie/kurs-arduino-w-robotyce-1-wstep-id936>

Podstawowe informacje na temat prądu elektrycznego:

<http://forbot.pl/blog/artykuly/podstawy/podstawy-elektroniki-1-napiecie-prad-opor-zasilanie-id3947>

Informacja o diodach LED:

https://pl.wikipedia.org/wiki/Dioda_elektroluminescencyjna

Jak działa płytka stykowa (prototypowa):

https://pl.wikipedia.org/wiki/P%C5%82ytka_prototypowa

Tworzenie gier:

https://pl.wikipedia.org/wiki/Tworzenie_gry_komputerowej

Działanie funkcji random() w Arduino:

<https://www.arduino.cc/en/reference/random>

Co to jest debugowanie:

<https://pl.wikipedia.org/wiki/Debugowanie>

Scenariusz został opracowany na potrzeby projektu „MoboLab – roboty i tablety w Twojej szkole”. Celem projektu jest zwiększenie kompetencji informatycznych z zakresu programowania i wykorzystywania technologii mobilnych w uczeniu się, a także kreatywności, innowacyjności i umiejętności współpracy w zespole z wykorzystaniem TIK, uczniów/uczennic z (UCZ) z 6 szkół podnadgimnazjalnych i 4 gimnazjów Wołomina i Zielonki. Projekt dofinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego (Regionalny Program Operacyjny Województwa Mazowieckiego na lata 2014-2020, Oś Priorytetowa X. Edukacja dla rozwoju regionu, Działanie 10.1. Edukacja ogólna i przedszkolna, Poddziałanie 10.1.2. Edukacja ogólna w ramach ZIT).



Ten utwór jest dostępny na licencji [Creative Commons Uznanie autorstwa 4.0 Międzynarodowe](https://creativecommons.org/licenses/by/4.0/).