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

SCENARIUSZ 1. WPROWADZENIE DO ARDUINO

scenariusz zajęć pozalekcyjnych

autor: Wojciech Karcz, Michał Podziomek

redakcja: Agnieszka Koszowska

SŁOWA KLUCZOWE:

Arduino, programowanie, mikrokontroler, physical computing, Blink, hello world

KRÓTKI OPIS ZAJĘĆ:

Podczas zajęć uczniowie i uczennice poznają projekt **Arduino**, dowiadują się, czym jest platforma Arduino, jak działa **mikrokontroler** i jak zaprogramować Arduino za pomocą komputera. Poznają i/lub utrwalają podstawowe pojęcia programistyczne (skrypt, program, algorytm, sterowanie, warunek, pętla). Na zakończenie zajęć wykonują zadanie: piszą swój pierwszy program pt. „Hello World”.

WIEDZA I UMIEJĘTNOŚCI ZDOBYTE PRZEZ UCZNIA / 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),
- zna podstawowe elementy interfejsu środowiska programistycznego Arduino IDE i podstawowe komendy języka Arduino IDE: **pinMode()**, **digitalWrite()**, **delay()**,
- zna podstawowe elementy języka **Scratch**, potrafi stworzyć prosty skrypt w tym języku.
- potrafi napisać prosty program w języku Arduino IDE: „Hello World”.

GRUPA DOCELOWA:

Starsze klasy szkoły podstawowej (VII-) i klasy gimnazjalne (po dostosowaniu: możliwość realizacji w młodszych klasach: I-III i IV-VI szkoły podstawowej).

LICZBA UCZNIÓW/UCZENNIC W KLASIE:

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”):

1

POTRZEBNY SPRZĘT I OPROGRAMOWANIE:

- komputer (przenośny lub stacjonarny),
- program Arduino IDE (do pobrania ze strony: http://www.arduino.org/downloads),
- (opcjonalnie) program Scratch for Arduino (do pobrania ze strony: http://s4a.cat/),
- płytka **Arduino UNO** i kabel **USB A-B** (dla każdego ucznia lub dla 2- lub 3-osobowego zespołu uczniów),
- projektor i laptop (w części teoretycznej).

CO NALEŻY PRZYGOTOWAĆ PRZED ZAJĘCIAMI:

- zainstalować program **Arduino IDE**,
- (opcjonalnie): zainstalować program **Scratch for Arduino**,
- sprawdzić, czy wszystkie komputery wykrywają podłączone Arduino,
- przeczytać dokładnie scenariusz,
- zapoznać się z mareriałami 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:

Wstęp - 20 minut

Podczas zajęć będziemy przekazywać uczniom podstawową wiedzę, która pozwoli im rozpocząć swoją przygodę z Arduino i zdobywać oraz rozwijać bardziej zaawansowane umiejętności programistyczne. Skupienie się na podstawowej wiedzy i jej utrwalenie jest na tym etapie niezwykle ważne i pozwoli uniknąć ewentualnych problemów czy frustracji, jakich mogą doświadczać uczniowie przy pracy nad kolejnymi projektami. Naszym głównym zadaniem jest dokładne wyjaśnienie uczniom, co to jest mikrokontroler, czym jest Arduino, jak podłączyć Arduino do komputera i je zaprogramować, czym jest program „Hello World" i jak działa on na Arduino.

Zajęcia rozpoczynamy od krótkiej dyskusji (10-15 min.) na temat automatów oraz urządzeń elektronicznych. Przykładowe pytania:

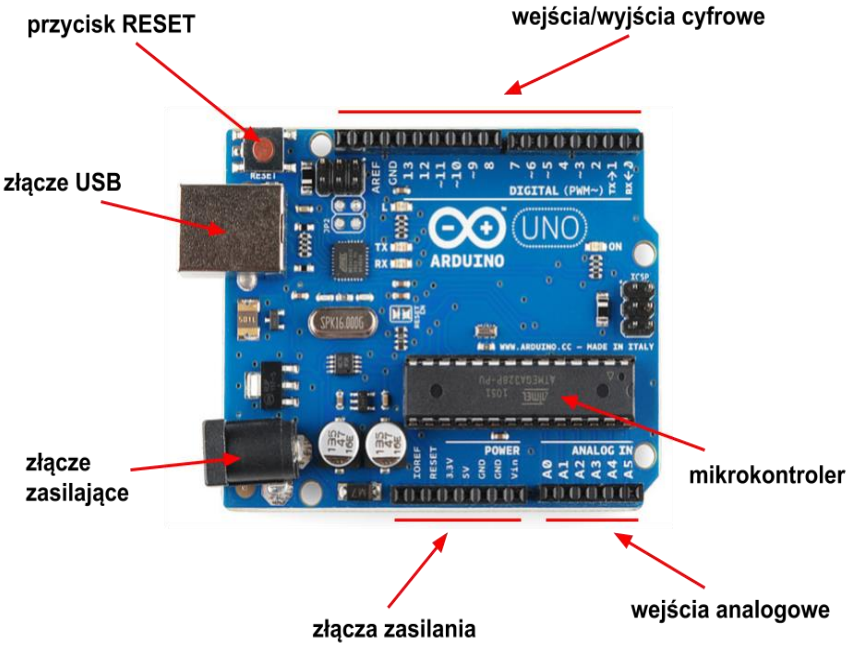
- *Co to są automaty? Jakie znacie przykłady?*
- *Jakie znacie urządzenia elektroniczne?*
- *Jak działają takie urządzania? Co robią? Jaka jest ich zasada działania?*
- *Co odpowiada za to, że latarnia miejska włącza się sama, kiedy jest ciemno?*
- *Dlaczego mówimy o „inteligentnych" urządzeniach? O tym, że maszyny „myślą"?*

Zbieramy odpowiedzi uczniów, a następnie opowiadamy o mikrokontrolerach. W prostych słowach tłumaczymy, czym są mikrokontrolery, jak działają oraz gdzie można je znaleźć. Warto też wspomnieć o „programowaniu fizycznego świata", czyli projektowaniu systemów interaktywnych, w których zachodzi wzajemne

oddziaływanie między człowiekiem i maszyną (tzw. HCI, czyli „human-computer interaction”).

Pokazujemy i/lub wymieniamy różne przykłady zastosowań mikrokontrolerów w codziennym życiu np. fotokomórka, pralka, wyłącznik czasowy, sygnalizacja świetlna, termostat, automatyczne drzwi rozsuwane itp. Następnie prezentujemy Arduino. Opowiadamy, że jest to przykład prostego mikrokontrolera, który można programować i na jego bazie samodzielnie projektować różne urządzenia, na przykład takie, które wcześniej wymieniliśmy. W tym momencie możemy również zapytać o pomysły uczniów na urządzenia, które chcieliby mieć.

Kolejnym krokiem jest wyjaśnienie budowy płytki Arduino UNO: pokazanie, z jakich elementów ona się składa – możemy wykorzystać poniższe zdjęcie lub wykonać własne:



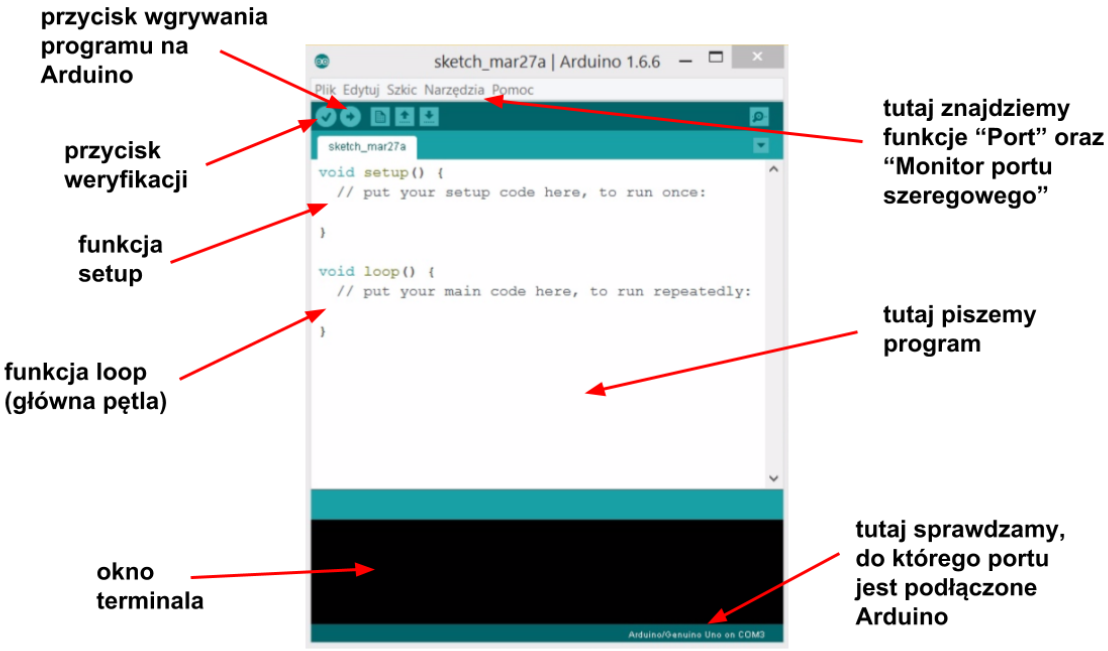
Przechodzimy do kolejnej części zajęć, w której pokażemy, jak podłączyć Arduino do komputera i sprawdzimy, jak ono działa. Wspominamy też, że celem naszych zajęć jest zapoznanie się z tym urządzeniem i nauczenie się, jak je obsługiwać.

Podłączenie Arduino do komputera - 10 minut

Arduino programujemy za pomocą komputera. A więc najpierw musimy podłączyć Arduino do komputera za pomocą kabla USB A-B. W języku i środowisku

programistycznym Arduino IDE tworzy się programy, które następnie wgrywa się (przesyła za pomocą kabla) na Arduino. Kiedy program zostanie wgrany, (trwa to kilku sekund), będzie on wykonywany bez przerwy (w pętli). Teraz pokazujemy uczniom, jak podłączyć Arduino do komputera.

Po podłączeniu uruchamiamy program Arduino IDE. Sprawdzamy, czy na każdym komputerze został poprawnie wykryty port, do którego jest podłączone Arduino (trzeba kliknąć w górnym menu w „Narzędzia", a następnie w „Port"). Uczulamy uczniów na to, że czasem mogą pojawiać się problemy z wykryciem Arduino na komputerze i warto wtedy sprawdzić, czy port jest aktywny. Następnie omawiamy wszystkie podstawowe funkcje Arduino IDE, takie jak przycisk weryfikacji, przycisk przesyłania programu, okno do pisania programu, miejsce, w którym pojawiają się komunikaty o błędach, miejsce, w którym możemy znaleźć gotowe przykłady itp.



Pierwszy program „Hello World". Cz. 1 - 15 minut

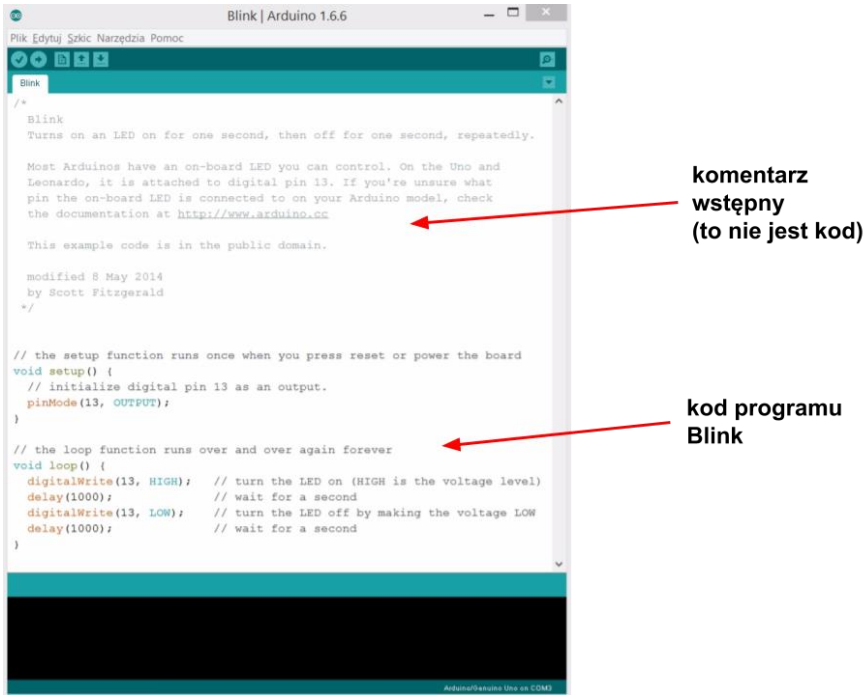
Po pokazaniu uczniom wszystkich podstawowych funkcji Arduino IDE przechodzimy do tworzenia i wgrywania naszego pierwszego programu, tzw. programu „Hello World". Rozpoczynamy krótką dyskusję dotyczącą programowania. Możemy wspomóc się następującymi pytaniami:

Co to jest program komputerowy? Jak on działa?

W jakim języku komunikujemy się z komputerem?
O jakich językach programowania słyszeliście?
W jaki sposób „myśli” komputer? Jak odczytuje napisany przez nas program?

Zbieramy wszystkie odpowiedzi i wyjaśniamy, jak działają programy komputerowe. Tłumaczymy też, że teraz będziemy uruchamiać pierwszy program na Arduino tzw. „Blink”, który jest programem typu „Hello World”. W tym momencie krótko przedstawiamy ideę programów typu Hello World (więcej informacji na ten temat znajduje się w materiałach dodatkowych – „Pigułce wiedzy i inspiracji”).

Następnie przechodzimy do zaprezentowania programu „Hello World” dla Arduino. Z menu wybieramy podstawowy przykład - klikamy: „Plik” > „Przykłady” > „01.Basics” > „Blink”. Weryfikujemy szkic i wgrywamy za pomocą przycisku ze strzałką na Arduino.



Zadajemy uczniom pytanie, czy zauważyli jakąś zmianę w działaniu Arduino? Powinni zauważyć migającą diodę oznaczoną literą L, przypisaną do pinu nr 13.

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

Pierwszy program „Hello World”. Cz. 2 - 20 minut

Rozpoczynamy zajęcia od krótkiego przypomnienia materiału z poprzednich zajęć i ponownego wgrania programu „Hello World” dla Arduino.

Analizujemy z uczniami cały kod programu odpowiedzialnego za miganie wbudowanej diody. Tłumaczymy strukturę każdego programu działającego na Arduino, składającego się z dwóch głównych funkcji: setup oraz void.

Omawiamy poszczególne funkcje tego programu:

pinMode() - funkcja odpowiedzialna za przypisanie odpowiedniego trybu (wejście/wyjście) do konkretnego pinu Arduino,

digitalWrite() - przypisywanie stanu niskiego/wysokiego do konkretnego pinu,

delay() - opóźnienie wyrażane w milisekundach.

W tym momencie odpowiadamy na ewentualne pytania uczniów i upewniamy się, czy wszyscy rozumieją działanie kodu. Dla przykładu zmieniamy wspólnie wartość jednej funkcji delay(), wgrywamy program na Arduino i patrzymy, jak zmienia się sposób migania diody.

Modyfikujemy program i wykonujemy zadanie – 25 minut

Na ostatnim etapie zajęć pozwalamy uczniom samodzielnie poeksperymentować z kodem i Arduino przez max. 10 min. W międzyczasie pomagamy wszystkim, którzy zgłaszają problemy. Na koniec przechodzimy do zadania końcowego opisanego poniżej (przewidywany czas: 10-15 min)

MOŻLIWE MODYFIKACJE DLA KLAS I-III I IV-VI:

Ten scenariusz pokazuje absolutne podstawy obsługi Arduino. Uczniowie z klas IV-VI powinni sobie poradzić z zadaniami opisanymi w scenariuszu – ewentualnie mogą potrzebować wsparcia ze strony osoby prowadzącej zajęcia.

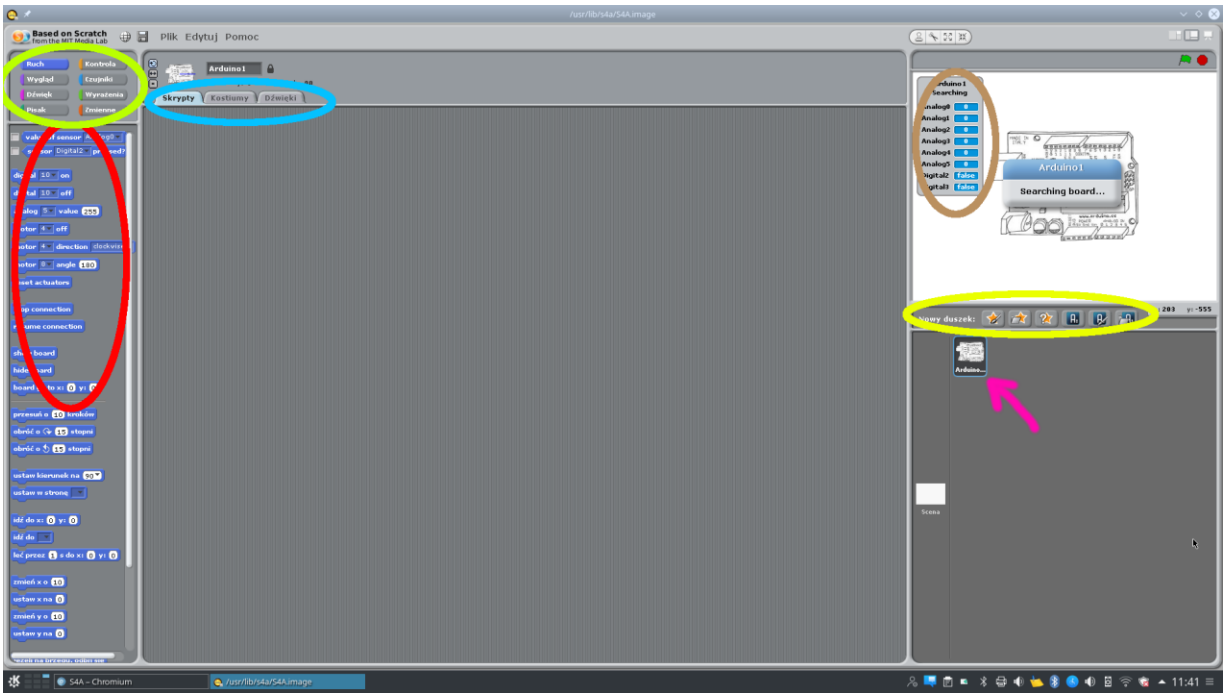
Pracując z uczniami w młodszych klasach można wykorzystać zamiast Arduino IDE program S4A (Arduino for Scratch) i w tym programie napisać pierwszy program „Blink". W klasach I-III można spróbować wykonać zadanie przy użyciu programu S4A (Arduino ror Scratch) lub mBlock. Uczniowie z klas IV-VI mogą wykonać zadania dodatkowe, jeżeli pozwoli na to czas i ich doświadczenie.

Wprowadzenie do programu S4A – Scratch for Arduino

Program Scratch for Arduino został stworzony po to, by można było łatwiej rozpocząć swoją przygodę z programowaniem. Program możemy pobrać ze strony: http://s4a.cat/.

Scratch for Arduino to modyfikacja oprogramowania Scratch (języka i środowiska programistycznego), przygotowana z myślą o osobach, które chcą w prosty sposób uczyć się podstaw programowania za pomocą Arduino i projektowania własnych urządzeń. Program jest oparty na środowsku Scratch, dlatego jego wygląd i funkcjonalności są podobne.

Na poniższym obrazku zostały zamieszczone kolorowe zaznaczenia, po to by precyzyjnie wyjaśnić podobieństwa i różnice.



Zieloną elipsą zostały zaznaczone kategorie bloków służących do budowania skryptów. Są one takie same jak w języku Scratch, z tą różnicą, że w kategorii „Ruch” jest dostępnych więcej bloków.

Bloki znajdujące się w czerwonej elipsie są blokami dedykowanymi dla Arduino, ale nimi zajmiemy się później. Niebieska elipsa oznacza zakładki aktywnego duszka, czyli skrypty, kostiumy i dźwięki. Żółta, natomiast, pokazuje opcje dodawania nowego duszka oraz nowego duszka Arduino.

Po prawej stronie na białej scenie znajduje się rysunek Arduino oraz komunikat „Arduino1 Searching board…”. Oznacza on, iż program S4A nie nawiązał połączenia z płytką. Aby nawiązać połączenie z Arduino, musimy wgrać na nie odpowiedni program, który będzie działał w sposób ciagły. Jego zadaniem jest tłumaczenie skryptów w języku Scratch (czyli bloków, które ułożyliśmy) na język zrozumiały dla płytki Arduino (czyli na odpowiedni kod). Taki program możemy pobrać z strony twórców programu S4A lub z innych miejsc w sieci – jest to program Open Source, dostępny na otwartych licencjach. Powstało też wiele modyfikacji programu, z których również można legalnie korzystać.

Tutaj można przeczytać o licencji MIT http://opensource.org/licenses/MIT.

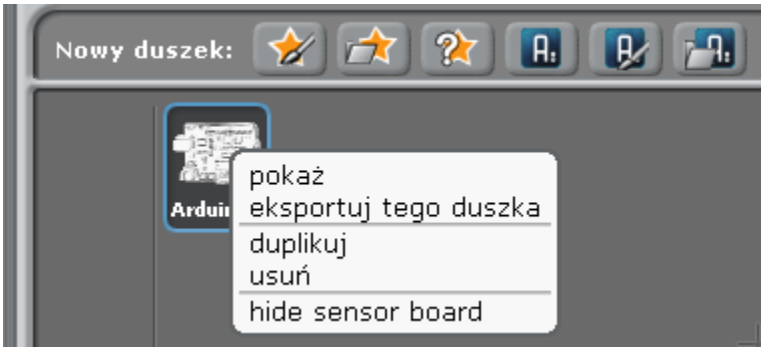
Program do pobrania ze strony twórców Scratch for Arduino:
http://vps34736.ovh.net/S4A/S4AFirmware16.ino

Program napisany przez blogera profesora Garcia, który uwzględnia działanie czujnika odległości
https://www.dropbox.com/s/czuegm2u5z22zvd/S4A_firmware_Profe_Garcia.txt?dl=0

Jest to zaawansowany program, dlatego nie będziemy go analizować, a ograniczymy się do wgrania go na płytkę za pomocą Arduino, tak samo jak wygrywaliśmy np. program Blink.

Gdy program będzie wgrany, diody RX i TX na Arduino powinny zacząć migać bardzo szybko, co świadczy o ciągłym przesyłaniu informacji między Arduino a S4A. Aby korzystać z S4A, najczęściej potrzebne jest odświeżenie programu S4A i przypomnienie mu, aby jeszcze raz sprawdził połączenie i poszukał płytki Arduino (dlatego przeprowadzamy taką operację).

W miejscu, w którym na rysunku zaznaczona jest różowa strzałka, znajdują się miniatury duszków. Gdy klikniemy prawym przyciskiem myszy duszka, który wygląda tak jak Arduino i nazywa się Arduino1, rozwinie się menu z opcjami. Wybieramy z niego opcję „usuń”. Wówczas nasz duszek zniknie.



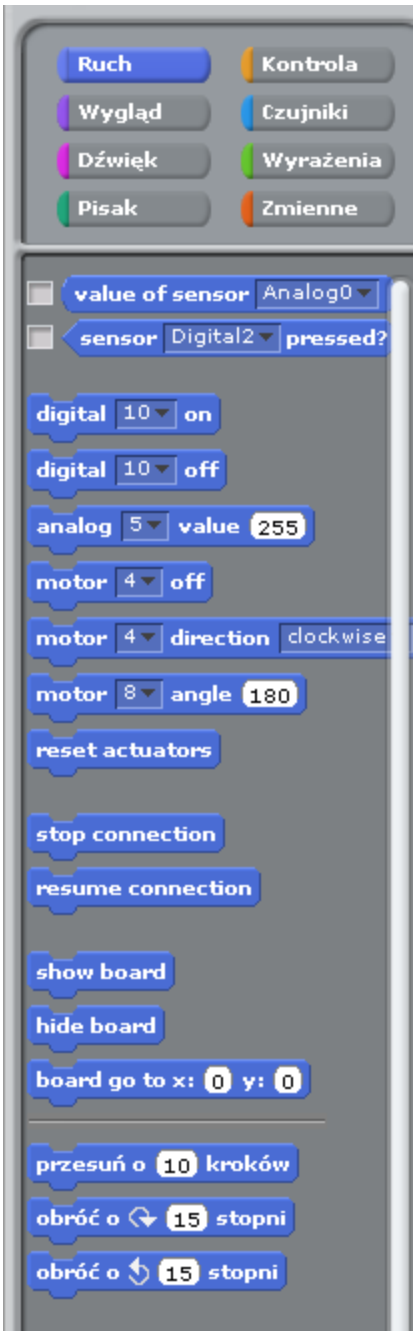
Dodajemy nowego duszka Arduino, klikając w ikonę z dużą literą „A”:

Połączenie powinno zostać nawiązane automatycznie, może się zdarzyć, że trzeba chwilę poczekać. Sprawdzamy, czy Arduino jest poprawnie podłączone obserwując poniższy rysunek znajdujący się przy duszku Arduino:

Rysunek przedstawia tablicę z wartościami odbieranymi z Arduino. Jeżeli te wartości się nie poruszają lub wynoszą zero, płytka nie jest podłączona poprawnie. Jeżeli wartości się zmieniają na dodatnie, oznacza to, że mamy połączenie.

Jeżeli nie ma połączenia, sprawdzamy, czy płytka jest podłączona do komputera - często pomaga wyjęcie i ponowne włożenie kabla USB.

Gdy połączenie jest poprawne, przechodzimy do omówienia pierwszych bloków z kategorii Ruch.

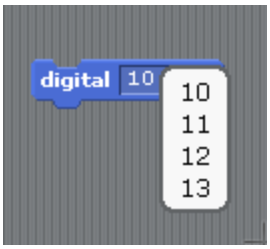

Ruch
Kontrola
Wygląd
Czujniki
Dźwięk
Wyrażenia
Pisak
Zmienne
value of sensor Analog0
sensor Digital2 pressed?
digital 10 on
digital 10 off
analog 5 value 255
motor 4 off
motor 4 direction clockwise
motor 8 angle 180
reset actuators
stop connection
resume connection
show board
hide board
board go to x: 0 y: 0
przesuń o 10 kroków
obróć o 15 stopni
obróć o 15 stopni

Omawiamy poszczególne bloki na przykładzie „digital… on” i „digital… off”.

Blok digital (10) ON (włączony) ustawia na Arduino stan wysoki, czyli wysyła napięcie 5V na pin o numerze 10.

Blok digital (10) OFF (wyłączony) ustawia na Arduino stan niski, czyli wysyła napięcie 0V na pin o numerze 10.

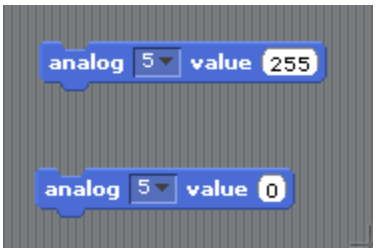
Piny wybieramy z listy rozwijanej, która otworzy się, gdy kliniemy strzałkę w polu z liczbą 10 – dostępne piny to: 10, 11, 12, 13.



Zadajemy uczniom pytanie: co stanie się, jeśli do pinu 10 zostanie podłączona dioda? I tworzymy skrypt jak na poniższym obrazku:

Po naciśnięciu flagi dioda zaświeci się (bo wysyłamy 5V stan wysoki na pin 10), świeci się przez 1 sekundę, po czym gaśnie, (bo wysyłamy 0V stan niski na pin 10).

Następnie omawiamy blok „analog… value…”. Wysyła on na pin 5 wartość od 0 do 255 (jest to proporcjonalne napięcie od 0V do 5V). Możemy wybrać inne piny niż 5 analogicznie jak w przypadku „digital… on” – piny wybieramy z listy rozwijanej, która pojawi się po kliknięciu strzałki. Dostępne wartości (piny) to: 5, 6, 9, 10.



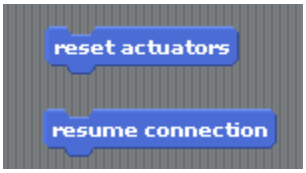
Te bloki możemy wykorzystać, jeśli chcemy w sposób płynny zmienić jasność świecącej diody, używając np. zmiennej czy precyzyjnego sterowania servem lub częstotliwością buzzera.

Jeśli podłączymy do Arduino silnik krokowy pod pin numer 4, to możemy nim sterować za pomocą poniższych bloków: motor… direction…” i „motor… off”.
Za pomocą bloku „motor ... direction…” ustawiamy kierunek ruchu silnika (zgodny z ruchem zegara lub przeciwny do ruchu zegara), a za pomocą „motor… off” wyłączamy silnik (czyli wysyłamy stan niski 0V).

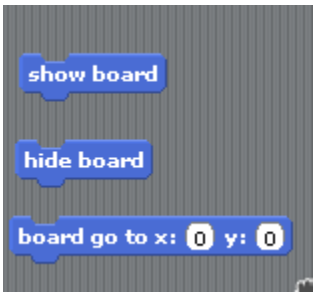
Następny blok „motor… angle…” jest przeznaczony do sterowania servem. Podpinamy sygnał serva pod pin numer 8. Angle oznacza tu kąt położenia orczyka serva.

W poniższym skrypcie po naciśnięciu flagi orczyk serva ustawi się pod kątem 0 stopni, poczeka 1 sekundę, a następnie ustawi orczyk serva pod kątem 180 stopni.

Dwa bloki „reset actuators” i „resume connection” odpowiadają za połączenie z Arduino: możemy je zresetować lub przywrócić. Te bloki są używane do bardziej zaawansowanych projektów.



Kolejne bloki: „show board”, „hide board” oraz „board go to x:… y:…” odpowiadają za wygląd i położenie duszka Arduino. Możemy duszka pokazać, ukryć lub ustawić jego dowolną pozycję na scenie za pomocą układu współrzędnych.



ZADANIE SPRAWDZAJĄCE UMIEJĘTNOŚCI ZDOBYTE PODCZAS ZAJĘĆ:

Uczeń / uczennica, pracując samodzielnie albo w dwu- lub trzyosobowym zespole tworzy skrypt, który będzie powodował, że dioda na Arduino przez cały czas miga, tzn. świeci się przez 2 sekundy, następnie gaśnie na 0,5 sekundy, znowu świeci się przez 2 sekundy, znowu gaśnie na 0,5 sekundy itd.

PIGUŁKA WIEDZY I INSPIRACJI DLA OSÓB PROWADZĄCYCH:

Kurs programowania Arduino Forbot:
http://forbot.pl/blog/artykuly/programowanie/kurs-arduino-w-robotyce-1-wstep-id936

Czym jest mikrokontroler:
https://pl.wikipedia.org/wiki/Mikrokontroler

O programowaniu fizycznego świata:
https://en.wikipedia.org/wiki/Physical_computing

Główna strona projektu Arduino:
https://www.arduino.cc/

Prezentacja twórcy Arduino o tym, jak powstawał projekt i jakie cele realizuje:
https://www.ted.com/talks/massimo_banzi_how_arduino_is_open_sourcing_imagination?language=pl

Czym jest program "Hello World":
https://pl.wikipedia.org/wiki/Hello_world

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).