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

SCENARIUSZ 27. CZUJNIK PARKOWANIA

scenariusz zajęć pozalekcyjnych

autor: Michał Kłosiński

redakcja: Agnieszka Koszowska

SŁOWA KLUCZOWE:

Arduino, programowanie, elektronika, czujnik odległości HC-SR04

KRÓTKI OPIS ZAJĘĆ:

Podczas zajęć uczniowie i uczennice uczą się, jak stworzyć układ elektroniczny za pomocą Arduino oraz **czujnika odległości HC-S04**. Przygotowują układ i tworzą skrypt w programie Scratch for Arduino pozwalający wykorzystanie czujnika odległości HC-S04 jako czujnika parkowania.

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),
- potrafi podłączyć czujnik odległości HC-SR04 do Arduino,
- wie, jak działa czujnik odległości HC-SR04,
- zna podstawowe elementy interfejsu programu Scratch for Arduino, wie, gdzie szukać potrzebnych bloków,
- 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).

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"):

1

POTRZEBNY SPRZĘT I OPROGRAMOWANIE:

- komputer (przenośny lub stacjonarny),
- program Arduino IDE (do pobrania ze strony: http://www.arduino.org/downloads),
- program Scratch for Arduino (do pobrania ze strony: http://s4a.cat/),
- płytka Arduino UNO i kabel USB A-B (dla każdego uczestnika lub dla pary uczestników),
- płytka stykowa,
- czujnik odległości HC-SR04,
- przewody połączeniowe,
- projektor i laptop (w części teoretycznej).

CO NALEŻY PRZYGOTOWAĆ PRZED ZAJĘCIAMI:

- zainstalować program Arduino IDE,
- zainstalować program 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.

Omawiamy zasadę działania czujnika odległości – 15 minut

Zapowiadamy, że na tych zajęciach będziemy korzystać z czujnika odległości. Krótko omawiamy, jak działa taki czujnik. Możemy zadać uczniom pytania:

- *czy wiecie, jak działa czujnik odległości?*
- *w jaki sposób „widzi" nietoperz?*

Ultradźwiękowy czujnik odległości, którym będziemy się posługiwać, działa analogicznie do echolokacji nietoperza. Wysyła sygnał i go odbiera, jeśli odbije się on od przeszkody. Na podstawie czasu odbioru informacji zwrotnej czujnik określa odległość od przeszkody.

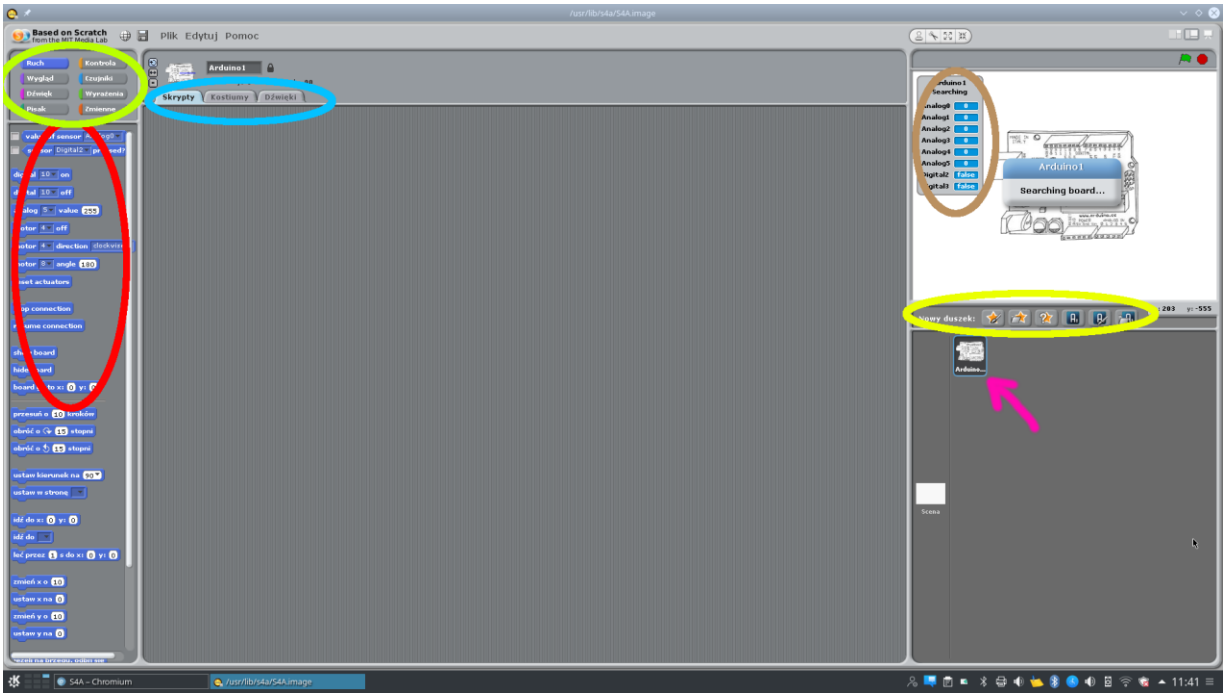
Ultradźwiękowy czujnik odległości HC-S04 pozwala mierzyć odległość w zakresie od 1 do 250 cm. Dzięki niemu możemy zbudować i zaprogramować robota, który będzie omijać przeszkody, czujnik parkowania czy instrument muzyczny.

Poznajemy (lub przypominamy sobie) program Scratch For Arduino – 10 minut

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

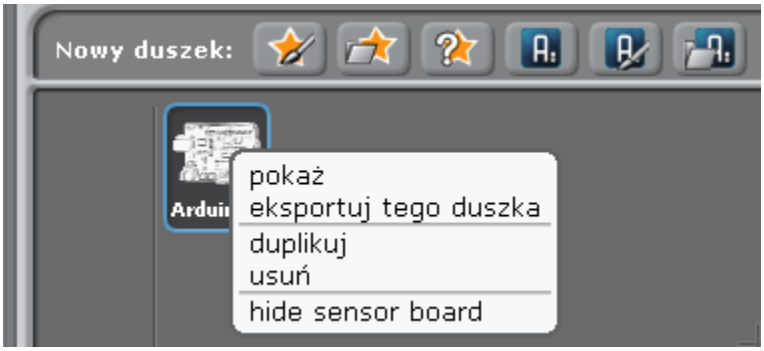
Program napisany przez blogera profesora Garcia, który uwzględnia działanie czujnika odległości:

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.

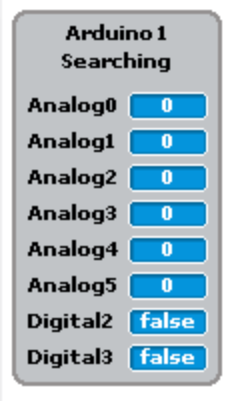
Gdy program będzie wgrany, diody RX i TX na Arduino powinny zacząć migać bardzo szybko, co świadczy o ciągłym przesyłaniu informacji między Arduino a S4A. Aby korzystać z S4A, najczęściej potrzebne jest odświeżenie programu S4A i przypomnienie mu, aby jeszcze raz sprawdził połączenie i poszukał płytki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

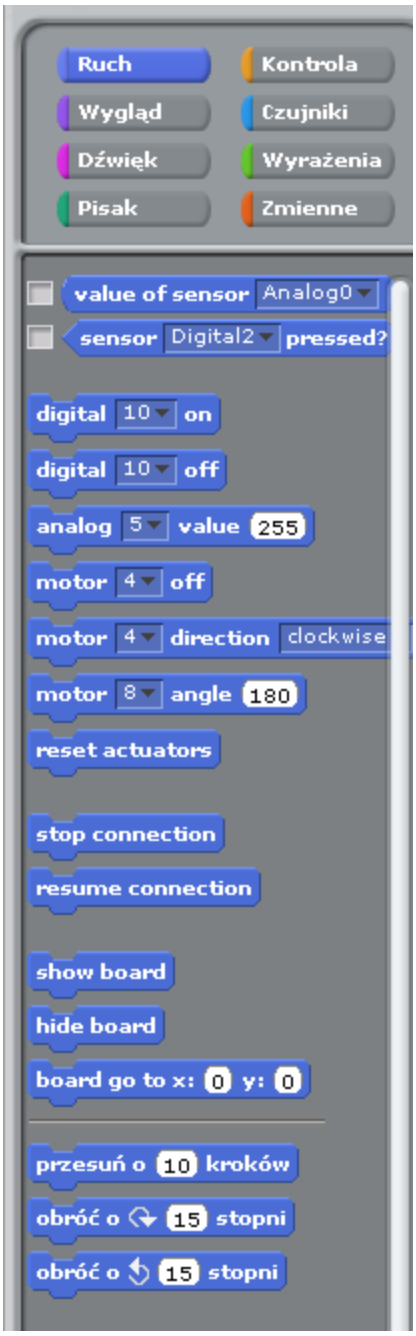
Połączenie powinno zostać nawiązane automatycznie, może się zdarzyć, że trzeba chwilę poczekać. Sprawdzamy, czy Arduino jest poprawnie podłączone obserwując poniższy rysunek znajdujący się przy duszku Arduino:



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.

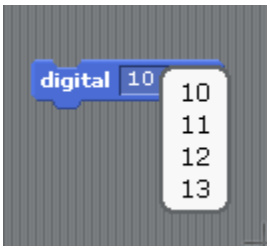

Ruch
Kontrola
Wygląd
Czujniki
Dźwięk
Wyrażenia
Pisak
Zmienne
value of sensor Analog0
sensor Digital2 pressed?
digital 10 on
digital 10 off
analog 5 value 255
motor 4 off
motor 4 direction clockwise
motor 8 angle 180
reset actuators
stop connection
resume connection
show board
hide board
board go to x: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

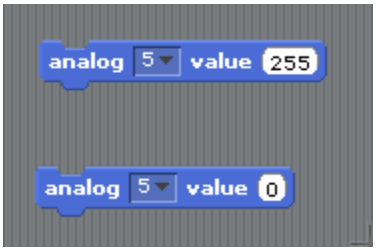
Piny wybieramy z listy rozwijanej, która otworzy się, gdy kliniemy strzałkę w polu z liczbą 10 – dostępne piny to: 10, 11, 12, 13.



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.



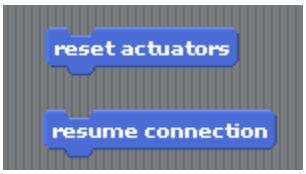
Te bloki możemy wykorzystać, jeśli chcemy w sposób płynny zmienić jasność świecącej diody, używając np. zmiennej czy precyzyjnego sterowania servem lub częstotliwością buzzera.

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).

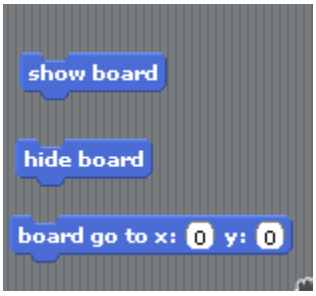
Następny blok „motor… angle…" jest przeznaczony do sterowania servem. Podpinamy sygnał serva pod pin numer 8. Angle oznacza tu kąt położenia orczyka serva.

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.



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

Przypomnienie materiału – 10 minut

Rozpoczynamy od krótkiego przypomnienia materiału z poprzedniej części zajęć i uruchomienia programu Scratch for Arduino.

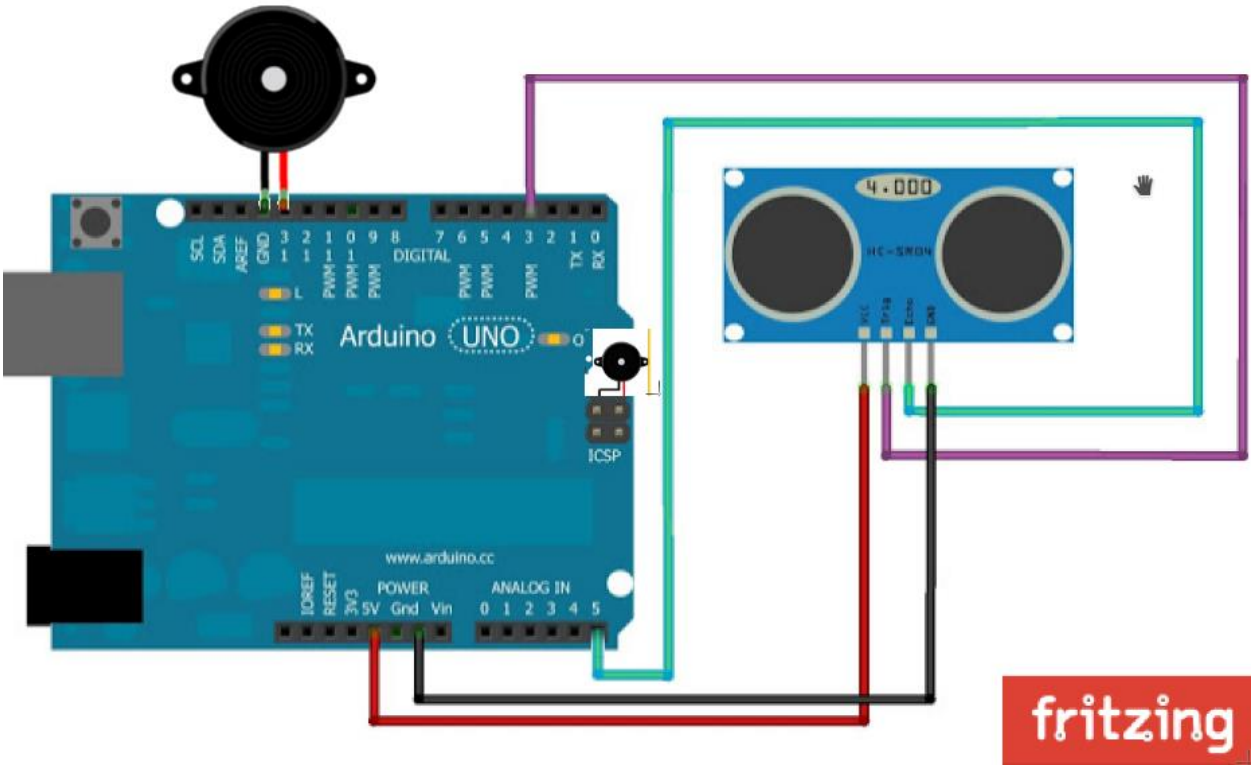
Montujemy układ – 10 minut

Podłączamy czujnik HC-SR04 do Arduino wg poniższego schematu.

Połączenia:

Arduino ---> HC-SR04
5V Vcc
3 Trig
A5 Echo
GND Gnd

Arduino 13 ---> buzzer (dłuższa nóżka)
Arduino GND-> buzzer (krótsza nóżka) GND



Piszemy skrypt wykorzystujący czujnik odległości jako czujnik parkowania – 25 minut

Uwaga! Na Arduino należy wgrać program odpowiedzialny za łączenie się programu Scratch For Arduino z czujnikiem odległości, autorstwa profesora Garcii: https://www.dropbox.com/s/czuegm2u5z22zvd/S4A_firmware_Profe_Garcia.txt?dl=0

Gdy wartości na tablicy sensor board się zmieniają i wiemy że Arduino jest połączone poprawnie, piszemy pierwszy skrypt dla wykorzystania czujnika odległości.

Podłączyliśmy echo z czujnika odległości do pinu A5 więc to tam będziemy nasłuchiwać informacji o napięciu, jakie dochodzą czujnika dlatego patrzymy na tablicę z wartościami na Analog5.

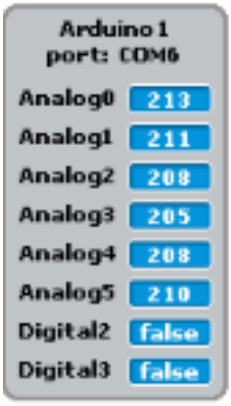
Możemy zbliżyć i oddalić rękę od czujnika odległości. Powinniśmy zaobserwować, że wartości się zmieniają w zależności od położenia naszej ręki od czujnika. Jednostki wyświetlające się w tablicy odpowiadają centymetrom.

Value of sensor to blok z zaokrąglonymi krawędziami, czyli blok zmieniających się w nim wartości i dlatego wiemy, że znajdują się w nim najczęściej liczby – w tym przypadku w centymetrach.

Wybieramy Analog5:

A więc:

value of sensor Analog5 to liczba 210.



Czyli w tym momencie czujnik znajduje się 210 cm od przeszkody. Dlatego teraz możemy uzależnić tą wartość w skryptach.

Wiedząc, jak daleko znajdujemy się od przeszkody, możemy zaprogramować czujnik parkowania.

Założenia projektu

Możemy zapytać uczniów i wspólnie na tablicy napisać, w jaki sposób ma zachowywać się Arduino oraz Scratch for Arduino. Albo narzucić założenia z góry. Na przykład:

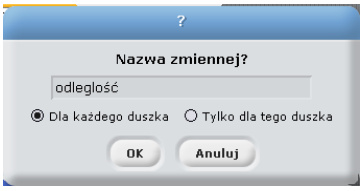
- jeżeli odległość od przeszkody jest większa niż 40, to buzzer „pika” raz na 2 sekundy, świadcząc o tym, że jesteśmy daleko od przeszkody,

- jeżeli odległość od przeszkody jest mniejsza niż 40, i jednocześnie większa od 10, to buzzer „pika” raz na 1 sekundę, świadcząc o tym, że jesteśmy już blisko przeszkody,

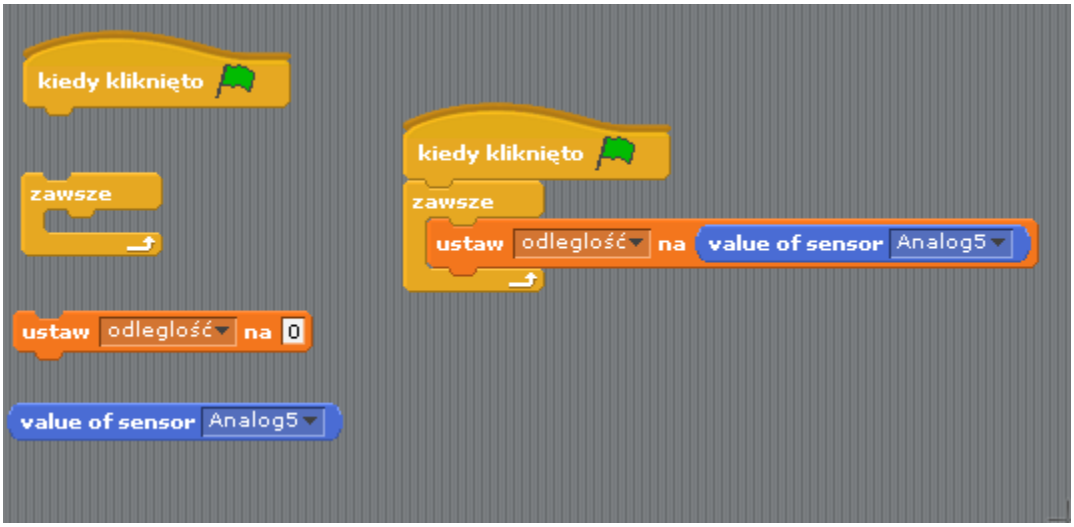
- jeżeli odległość od przeszkody jest mniejsza niż 10, to buzzer „pika” raz na 0,3 sekundy, świadcząc o tym, że jesteśmy już prawie na przeszkodzie.

Korzystamy z zmiennych. Wchodzimy w kategorię **Zmienne**. Dodajemy zmienną o dowolnej nazwie pasującej do zadania, np. zmienną **odległość**.

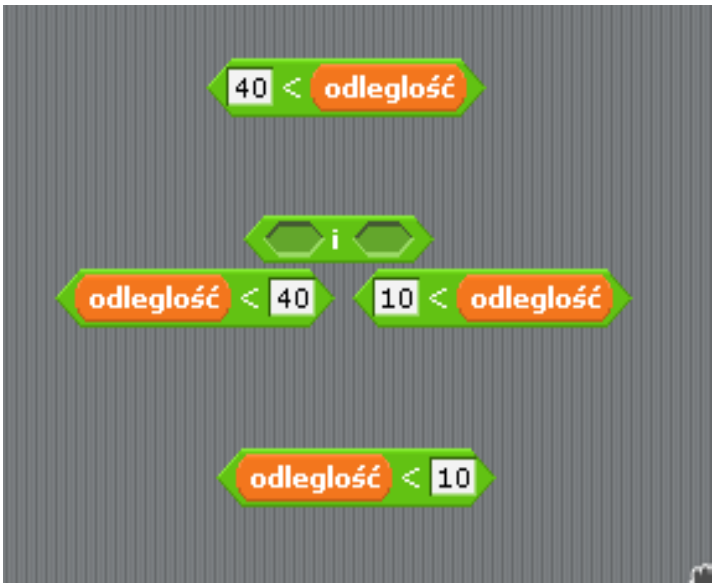
Program zapyta nas, czy ta zmienna ma być dla 1 konkretnego duszka, czy dla wszystkich duszków. Właśnie dzięki temu, że możemy zaznaczyć **dla wszystkich duszków**, możemy tworzyć interakcje między duszkiem Arduino a klasycznym duszkiem z obrazkiem.



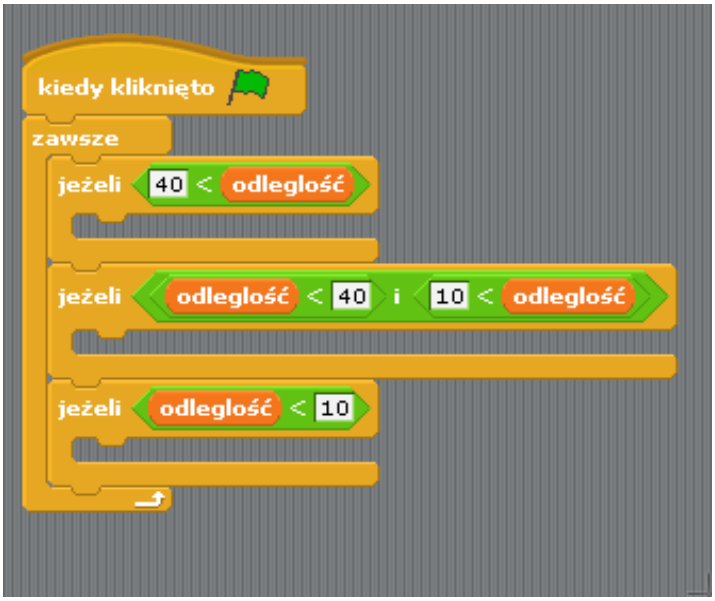
Ustawiamy zmienną **odległość**, tak aby była równa wartości z value of sensor Analog5 i zapętlamy ją, aby zawsze była aktualnie pobierana z Arduino.
Ten skrypt układamy w duszku **Arduino**.



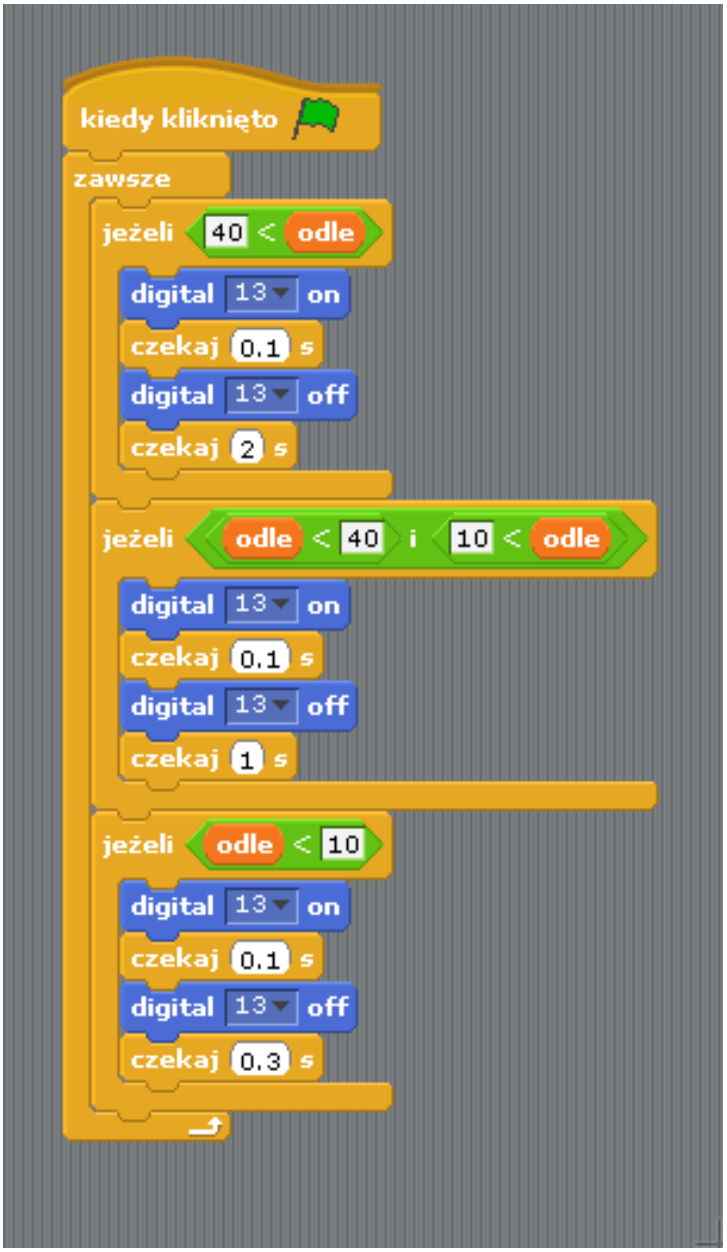
Wybieramy i ustalamy bloki odpowiadające za sprawdzanie warunku.
Oczywiście zmienna **odległość** jest równa = value of sensor Analog5. Moglibyśmy ją pominąć i wszędzie w warunki włożyć blok value of sensor Analog5. Ale czytelne ustawienie zmiennych to dobra praktyka, która daje nam możliwość rozwijania tego projektu na inne duszki.



Ustawiamy blok startowy w postaci **kiedy kliknięto flagę** – **zawsze** „sprawdzaj warunki”.



I teraz czas dodać instrukcje związane z uruchamianiem i częstotliwością buzzera.



I to gotowy czujnik parkowania, który możemy zastosować np. w samochodzie. Możemy też, jak w każdym projekcie, pozmieniać wartości, tak aby bardziej odzwierciedlały rzeczywistość.

Zadanie/wyzwanie 1

Należy dodać do czujnika sygnalizację świetlną, np. w postaci podłączenia 3 diod lub jednej RGB, która sygnalizuje odległość.

Zadanie/wyzwanie 2

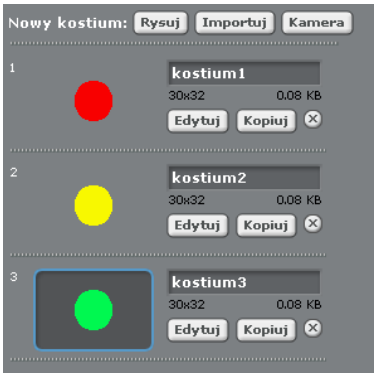
Należy dodać do czujnika sygnalizację świetlną, np. w postaci duszka i 3 kostiumów.

Możemy w zależności od czasu, który pozostał do końca zajęć, przejść przez zadania krok po kroku.

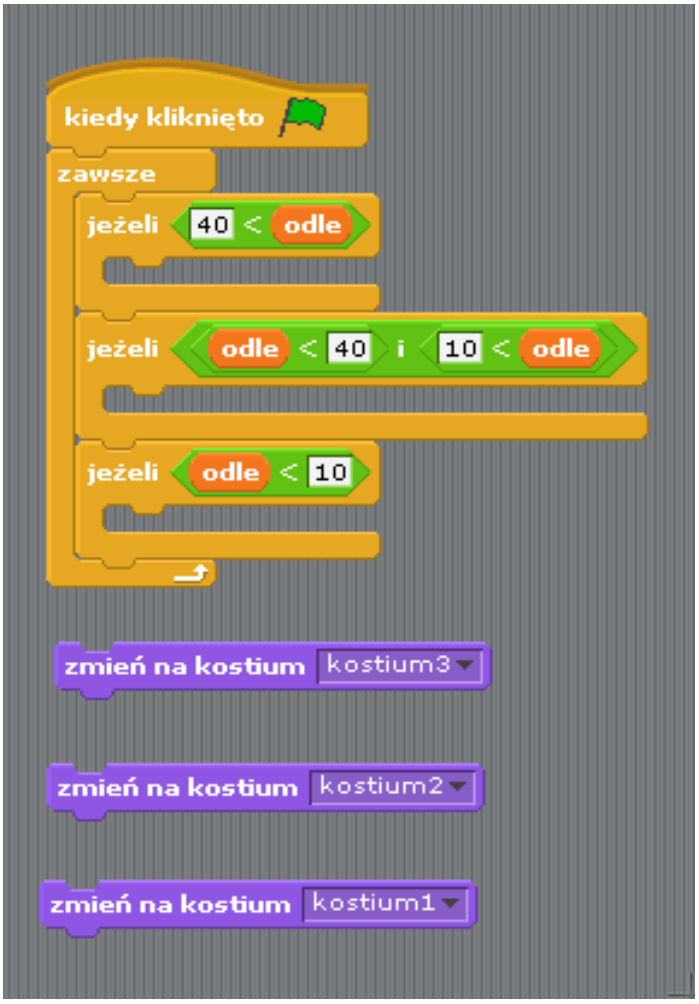
Rozwiązanie zadania 2

Tworzymy nowego duszka graficznego.

Nadajemy mu 3 różne kostiumy.

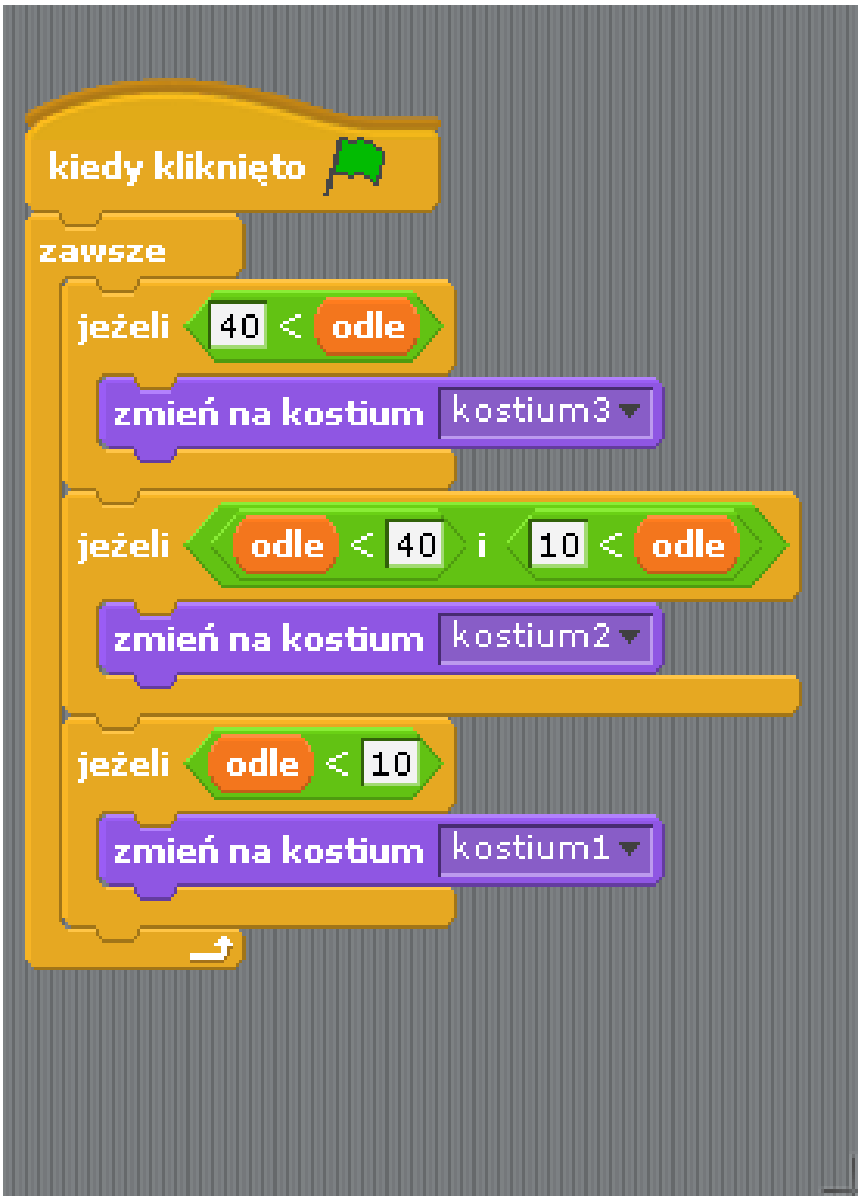


Uzależniamy kostiumy od warunków.

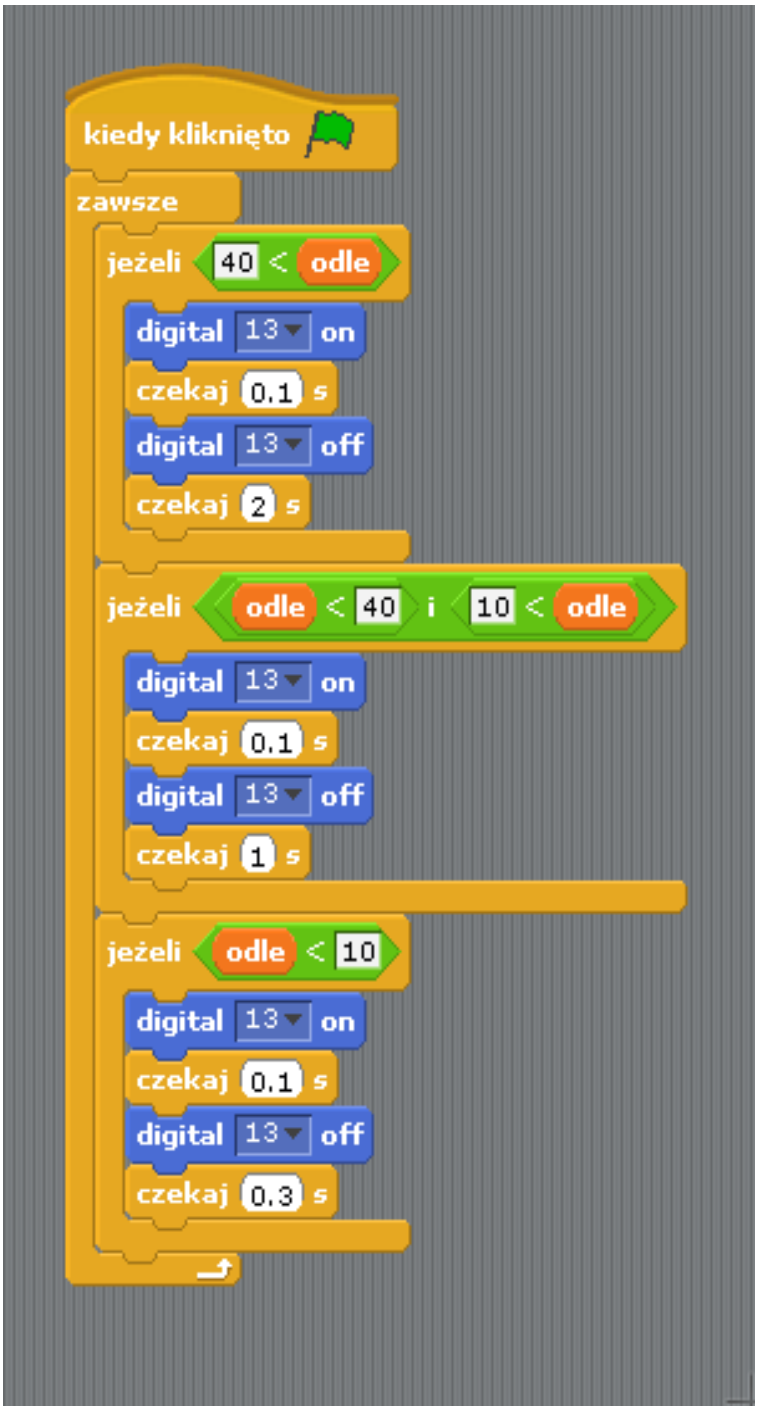


I teraz jednocześnie gdy klikniemy zieloną flagę nad sceną uruchomią się wszystkie skrypty.

Skrypty w duszku z kostiumami.



Oraz skrypty z Arduino



I jednocześnie zmieniając wartości na czujniku odległości, czyli zbliżając się i oddalając, będziemy obserwowali reakcje w postaci dźwięku i zmiany kostiumów.

MOŻLIWE MODYFIKACJE DLA MŁODSZYCH KLAS:

Ten scenariusz można zrealizować z uczniami w młodszych klasach, pracując w programie Scratch for Arduino. 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 buduje układ z wykorzystaniem Arduino i czujnika odległości HC-SR04. Zadanie polega na podłączeniu czujnika odległości, wytłumaczeniu, gdzie biegną piny zasilające i sygnałowe, oraz wyjaśnieniu zasady działania czujnika parkowania i modyfikacji w programie.

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

Jak działa płytka stykowa (prototypowa):
https://pl.wikipedia.org/wiki/P%C5%82ytka_prototypowa

Czym jest echolokacja?
https://pl.wikipedia.org/wiki/Echolokacja

Strona programu Arduino For Scratch
http://s4a.cat/

Zasady bezpieczeństwa w postępowaniu z modułami Arduino:
https://www.rugged-circuits.com/10-ways-to-destroy-an-arduino/

Różne sposoby wykorzystania czujnika odległości:
http://www.instructables.com/howto/HC-SR04/

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).