



Scenariusz zajęć

Automatyka w miniaturze

Wszystkie materiały i dane kontaktowe znajdują się na stronach internetowych projektów oraz w profilu projektu Erasmus+:
<https://sites.google.com/campus.ul.pt/hands-on-remote-language/home>
<https://erasmus-plus.ec.europa.eu/projects/search/details/2020-1-DE02-KA226-VET-008295>

Rozwój zespołu głównego

- Marion Pellowski i Lorenz Kampschulte, Deutsches Museum, Monachium, Niemcy
- Pedro Reis, Mónica Baptista, Luís Alexandre da Fonseca Tinoca, Uniwersytet Lizboński, Instytut Edukacji, Lizbona, Portugalia
- Wojciech Karcz, Adam Zahler, Anna Strzeszewska-Potyrała, Karolina Klimaszewska, Centrum Nauki Kopernik, Warszawa, Polska
- Miriam Voß, Mike Kramler, Marion Pellowski, Uniwersytet Techniczny w Monachium, Monachium, Niemcy

Oświadczenie

Ten projekt został zrealizowany przy wsparciu finansowym Komisji Europejskiej. Publikacja odzwierciedla jedynie stanowisko autora i Komisja Europejska nie ponosi odpowiedzialności za umieszczoną w niej zawartość merytoryczną.



Co-funded by the
Erasmus+ Programme
of the European Union

Wydawca

Autorzy:

Marion Pellowski, Uniwersytet Techniczny w Monachium i Deutsches Museum,
Mike Kramler, Miriam Voß, Uniwersytet Techniczny w Monachium,
obie instytucje Monachium, Niemcy.

Wersja polska przetłumaczona i zaadaptowana przez: Wojciech Karcz, Adam Zahler,
Anna Strzeszewska-Potyrała, Karolina Klimaszewska, Centrum Nauki Kopernik,
Warszawa, Polska

Muzeum Niemieckie, Monachium, Niemcy

Układ i projekt: Michał Romański

Druk: luty 2023 r.



Ta praca jest objęta licencją Creative Commons Uznanie autorstwa 4.0 International License. Elementy oznaczone jako cytaty mogą podlegać innym licencjom.

Treść

Podziękowanie.....	2
1 Wstęp.....	3
Krótki przegląd modułu	3
1.1 Cele.....	4
1.2 Przesłanki dydaktyczne.....	5
1.3 Duch społeczności i współpraca cyfrowa.....	7
1.4 Krótkie podsumowanie.....	8
2 Powiązanie z programem nauczania.....	11
3 Przegląd sekwencji modułów dla modułu "Automatyka w miniatu- rze".....	12
4 Treść: Kolejność uczenia się dla modułu nauczania "Automatyka w miniatu- rze".....	13
4.1 Jednostka 1.a: Wprowadzenie do tematu	13
4.2 Jednostka 1.b: Wprowadzenie do podstawowych obwodów i programowania Arduino	15
4.3 Jednostka 2.a: Symulacja online.....	18
4.4 Dział 2.b: Serwomotor.....	19
4.5 Jednostka 3.a: Obrotnica	21
4.6 Jednostka 3.b: Bariera świetlna	23
4.7 Jednostka 4: Rozszerzenia procesu automatyzacji	25
4.8 Jednostka 5: Współpraca uczniów.....	27
4.9 Jednostka 6: Współpraca cyfrowa	28
4.10 Jednostka 7: Automatyka w kontekście społecznym.....	29
4.11 Jednostka 8: Inne zastosowania stolika obrotowego	32
4.12 Ogólne uwagi dotyczące wideokonferencji i pracy praktycznej w sytuacjach na odległość.....	34
5 Możliwości zastosowania modułowego	36
Procedura 1.....	36
Procedura 2	36
6 Dodatek: Wgląd w interaktywną instrukcję multimedialną strona	37

Podziękowanie

Chcielibyśmy serdecznie podziękować nauczycielom Miejskiej Szkoły Zawodowej Informatyki i Integracji Systemów Informatycznych w Monachium za współpracę i rady oraz za stałe i zawsze konstruktywne wsparcie, które było niezwykle pomocne przy tworzeniu tego modułu nauczania. Dziękujemy młodym ludziom, którzy testowali z nami moduł, za ich opinie i wkład, który został uwzględniony w optymalizacji modułu. Chcielibyśmy również podziękować wszystkim nauczycielom, którzy uczestniczyli w naszej analizie potrzeb, za ich bardzo cenny wkład i uwagi dotyczące tego projektu.

Krótki przegląd modułu

Moduł dydaktyczny
**Automatyka
w miniaturze**

Uczniowie programują mały zakład produkcyjny i wprawiają go w ruch. Automatyczny system napędzania jest wykonany z prostych materiałów i może być zabrany przez uczniów do domu. Uczniowie mogą pracować w zespole, nawet w odległych sytuacjach.

Główny moduł nauczania			Jednostki dodatkowe / opcjonalne	
Jednostka 1a Krótkie wprowadzenie do tematu Cel: Tworzenie odniesień do życia codziennego i środowisk zawodowych Zadania: Pokaz filmów przedstawiających różne zastosowania stolików obrotowych w automatyce	Jednostka 2a ↔ Jednostka 3a Symulacja online Cel: Poznanie środowiska symulacyjnego online „Wokwi”, które umożliwia wymianę kodu nawet w nauczaniu na odległość Zadanie: Znaleźnię błędów w pliku Wokwi-debugging i obserwacja efektów na symulowanych diodach LED	Jednostka 3a Stolik obrotowy Cel: Budowa centralnej części zakładu produkcyjnego Zadanie: Ustawienie stolika obrotowego, wyregulowanie silnika napędowego i wpuszczenie stolika w ruch o ćwierć obrotu	Jednostka 4 Rozszerzenia procesu automatyzacji Cel: Przeniesienie wcześniej zdobytej wiedzy na nowe elementy, przestrzeń dla własnych projektów uczniów Zadania: Swobodny wybór w zakresie montażu innych elementów funkcjonalnych takich jak <ul style="list-style-type: none"> • Wskaźnik statusu LED • Zjeżdżalnia wejściowa • Wyrzutnia. 	Jednostka 5 / 6 Współpraca / Współpraca cyfrowa Cel: Promowanie współpracy między uczniami, w tym pracy zespołowej i poczucia wspólnoty nawet w przypadku kształcenia na odległość Zadania: Sprzężenie stolików różnych zespołów; Wykorzystanie transmisji światła do wywoływania efektów w sytuacjach odległych.
Jednostka 1b Wprowadzenie do Arduino Cel: Wprowadzenie do programowania za pomocą Arduino IDE, z wykorzystaniem pierwszego projektu obwodu. Zadania: Programowanie różnokolorowych diod LED, nauka podstawowych poleceń i częstych błędów w programowaniu	Jednostka 2b Serwomotor Cel: Poznanie właściwości i funkcji serwowatorów Zadanie: Zastosowanie serwowatorów do przemieszczania elementów funkcjonalnych zakładu produkcyjnego	Jednostka 3b Bariera świetlna Cel: Wykorzystanie czujników w zautomatyzowanej produkcji, omówienie pomiarów za pomocą czujników Zadanie: Zastosowanie fotorezystora (LDR) jako czujnika w fotokomórcie refleksyjnej do sterowania procesem produkcyjnym	Jednostka 7 Automatyka w kontekście społecznym Cel: Poznanie i omówienie wpływu procesów automatyzacji na społeczeństwo Zadanie: Omówienie zalet i wad automatyzacji z wykorzystaniem tablicy online oraz uwzględnienie takich aspektów jak zapotrzebowanie na energię, odpady itp.	Jednostka 8 Inne zastosowania stolika obrotowego Cel: Zachęcenie do dalszego zaangażowania się w pracę ze stolikiem obrotowym Zadania: Wykorzystanie stolika obrotowego jako sortownika wykrywającego różnice w jasności lub kolorze; Wykorzystanie stolika obrotowego jako odtwarzacza dźwięku wywołanego światłem, pomiar jasności



Nauczyciele mogą zmienić kolejność jednostek i rozpocząć od 3a/3b i kontynuować od 2a/2b.



W tym module mini linia produkcyjna pozwala uczniom zaprogramować prawdziwy system mechaniczny. Młodzi ludzie włączają efekty i rozwiązują problemy, które nie występują w środowisku czysto wirtualnym. Aby mogli zaprojektować działający proces produkcyjny, interakcja pomiędzy programowaniem a mechanicznymi częściami funkcjonalnymi mini zakładu produkcyjnego jest centralna. Kluczową ideą tego modułu dla nauczania na odległość jest zastąpienie wysokiej jakości i drogiego produktu dla klasy prostszą i tańszą wersją, którą nauczyciele mogą dać swoim uczniom do zabrania do domu.

Młodzież poznaje budowę i funkcjonowanie złożonego systemu technicznego na przykładzie mini zakładu produkcyjnego. Sterując serwomotorami jako siłownikami i wykorzystując czujnik światła do kontroli procesu napełniania, programują i sterują małym zakładem produkcyjnym. Programują tekstowy mikrokontroler Arduino, za pomocą którego mogą również rejestrować mierzone wartości. Uczniowie testują swoje programy na rzeczywistym przykładzie mini systemu automatyzacji i stosują w swoich testach metody rozwiązywania problemów. Muszą również uwzględnić wpływy środowiska (np. wpływ światła otoczenia na wartości pomiarowe czujnika światła).

Nauczyciele mogą wybrać jeden z dwóch różnych poziomów programowania i w ten sposób dostosować moduł do różnych poziomów wcześniejszej wiedzy: poziom podstawowy, który umożliwia obsługę minisystemu produkcyjnego za pomocą zaledwie kilku, prostych linii kodu – oraz poziom zaawansowany, który umożliwia wprowadzenie do programowania obiektowego (definicja obiektów, struktura obiektów, wykorzystanie praktycznych klas i obiektów). Nauczyciele mogą również wykorzystać moduł do przeniesienia istniejącej wiedzy z zakresu programowania obiektowego na przykład aplikacyjny. Jednak zaprogramowanie całego procesu automatyzacji jest również bardzo możliwe bez znajomości programowania obiektowego.

Automatyzacja i jej społeczne konsekwencje dotyczą wielu uczniów szkół zawodowych bezpośrednio, w ramach ich przyszłych obszarów zawodowych. Na ostatniej lekcji uczniowie są zachęceni do dyskusji i refleksji nad społecznymi kwestiami związanymi z automatyzacją, aby rozważyć własną przyszłość zawodową w szerszym kontekście.

Ze względu na bardzo istotny temat społeczny, młodzież zajmuje się różnymi zaletami i wadami automatyzacji dla naszego społeczeństwa, w tym aspektami ekologicznymi i związanymi z pracą.

1.2 Przesłanki dydaktyczne



Projekt wykorzystuje łatwe w użyciu narzędzia internetowe, takie jak środowisko symulacji online i tablica online, aby ułatwić nauczanie na odległość. Nauczyciele mogą korzystać z tych narzędzi online, jak również z praktycznych eksperymentów zarówno w nauczaniu bezpośrednim, jak i na odległość. W ten sposób można stworzyć wspólną płaszczyznę dla różnych i zmieniających się sytuacji dydaktycznych.

W przypadku wprowadzenia do tematu i pracy z miniprodukcją wideokonferencja jest odpowiednią i ważną metodą, zwłaszcza na początku. W dalszej części kursu wideokonferencje pozostają ważne, aby utrzymać kontakt z uczniami i wspierać ich w pracy. Możliwe są różne formy: wideokonferencje z całą klasą, wideokonferencje z podgrupami w salach i pokój wsparcia, w którym nauczyciel jest dostępny w razie pytań w określonym czasie. Moduł wykorzystuje również wideokonferencje do wspierania współpracy między studentami (patrz 1.3).

Po wprowadzeniu do podstaw młodzież może kontynuować samodzielną pracę z pomocą multimedialnej strony instruktażowej. Mogą sami zdecydować, jak chcą postępować. Rolą nauczyciela jest bycie osobą kontaktową w przypadku ewentualnych problemów. Wykorzystanie multimedialnej strony instruktażowej i samodzielna praca uczniów ułatwiają procedurę zarówno w sytuacjach bezpośrednich, jak i zdalnych. Połączenie narzędzi cyfrowych, tj. narzędzi online, różnych form wideokonferencji i wskazówek multimedialnych, pozwala na wykorzystanie modułu do nauczania na odległość, jak i hybrydowego. Nauczyciele pełnią w tym procesie rolę moderatorów procesu uczenia się, czy to na odległość, czy twarzą w twarz.

Automatyka jest ważnym elementem współczesnych społeczeństw i determinuje dużą część naszego życia. Łatwo zatem o odniesienia do rzeczywistych procesów automatyzacji i do codziennego życia uczniów. Dzięki procesom automatyzacji możliwe są odniesienia do różnych dziedzin zawodowych. Niezbędne procesy programowania i sterowania są częścią programu nauczania różnych zawodów szkoleniowych. W porównaniu z przemysłowymi urządzeniami

produkcyjnymi dla szkół, miniprodukcja składa się z prostych materiałów, w których poszczególne części można stosunkowo łatwo wymienić. Ma to istotne znaczenie dla udostępnienia zestawów materiałów uczniom w ramach kształcenia na odległość. Ponadto ważne wydawało się atrakcyjne zaprojektowanie zestawu, aby otworzyć możliwość, że młodzi ludzie będą z przyjemnością pracować z nim także w wolnym czasie.

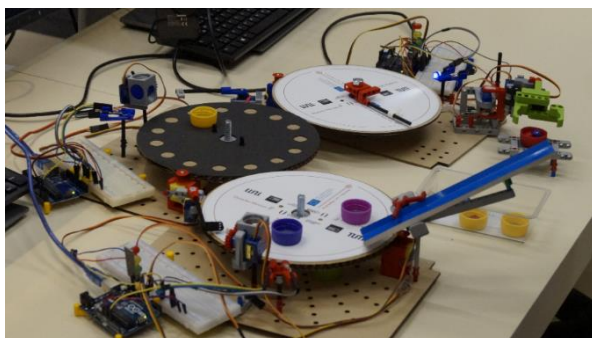
Elastyczne wykorzystanie mini zakładu produkcyjnego lub jego przeprojektowanie do innych zastosowań pozwala na dostosowanie stopnia złożoności do danej grupy docelowej – a tym samym na wykorzystanie zakładu w bardzo niejednorodnym kontekście europejskim.

Więcej szczegółów na temat opracowania i realizacji znajduje się w dołączonej broszurze "Przewodnik 2023". Zgodnie z kategoryzacją zawartą w przewodniku (patrz rozdział 2), ten moduł nauczania łączy eksperymenty lokalne i rzeczywiste (wymagające np. wysyłki lub zbierania materiałów) z eksperymentami zdalnymi i wirtualnymi, które ze względu na swoją strukturę mogą stanowić lepszą podstawę do zrozumienia koncepcyjnego.



Wykonywanie praktycznych eksperymentów w ramach pracy zespołowej ma ogromny potencjał w zakresie rozwijania poczucia wspólnoty wśród uczniów. Niektóre z zadań w tym module są zatem skoncentrowane głównie na uczeniu się we współpracy.

Obejmuje to w szczególności zadanie połączenia różnych miniprodukcji i ich wspólnej pracy.



Młodzież może łączyć swoje stoliki obrotowe, używając wyrzutnika do przenoszenia kubka z jednego stolika na drugi.

Takie sprzężenie stolików jest możliwe nawet wtedy, gdy żaden z zespołów nie zaprogramował jeszcze kompletnego procesu automatyzacji. W takim przypadku studenci po prostu używają innej części funkcjonalnej na każdym ze stolików i ustanawiają wspólny mechanizm transferu między różnymi stolikami.

Dzięki środowisku symulacyjnemu online uczniowie mogą wymieniać swoje programy w bardzo bezpośredni sposób i wspólnie pracować nad zadaniami programistycznymi. W ten sposób ścisła współpraca pozostaje możliwa nawet w nauczaniu na odległość.



Moduł ten zawiera również niezwykle możliwość współpracy na odległość: młodzi ludzie mogą współdziałać ze sobą w wideokonferencjach poprzez transmisję światła. Transmisja światła umożliwia przenoszenie obiektów fizycznych w inne miejsce, nawet bardzo odległe. Transmisja światła działa w następujący sposób: Partner pierwszego zespołu zapala – np. w Monachium – diodę LED na kamerze internetowej do wideokonferencji. Przed laptopem drugiego partnera zespołowego zamontowany jest czujnik światła, aby mógł on wykryć ten sygnał. W odpowiedzi na sygnał świetlny w lokalizacji drugiego partnera zespołowego, np. w Lizbonie, obracany jest serwomotor, który porusza funkcjonalną część zakładu produkcyjnego. Następnie drugi partner zespołu może z kolei przekazać sygnał świetlny i tak dalej. Celem tej formy współpracy jest wykorzystanie motywacyjnych zalet pracy zespołowej w

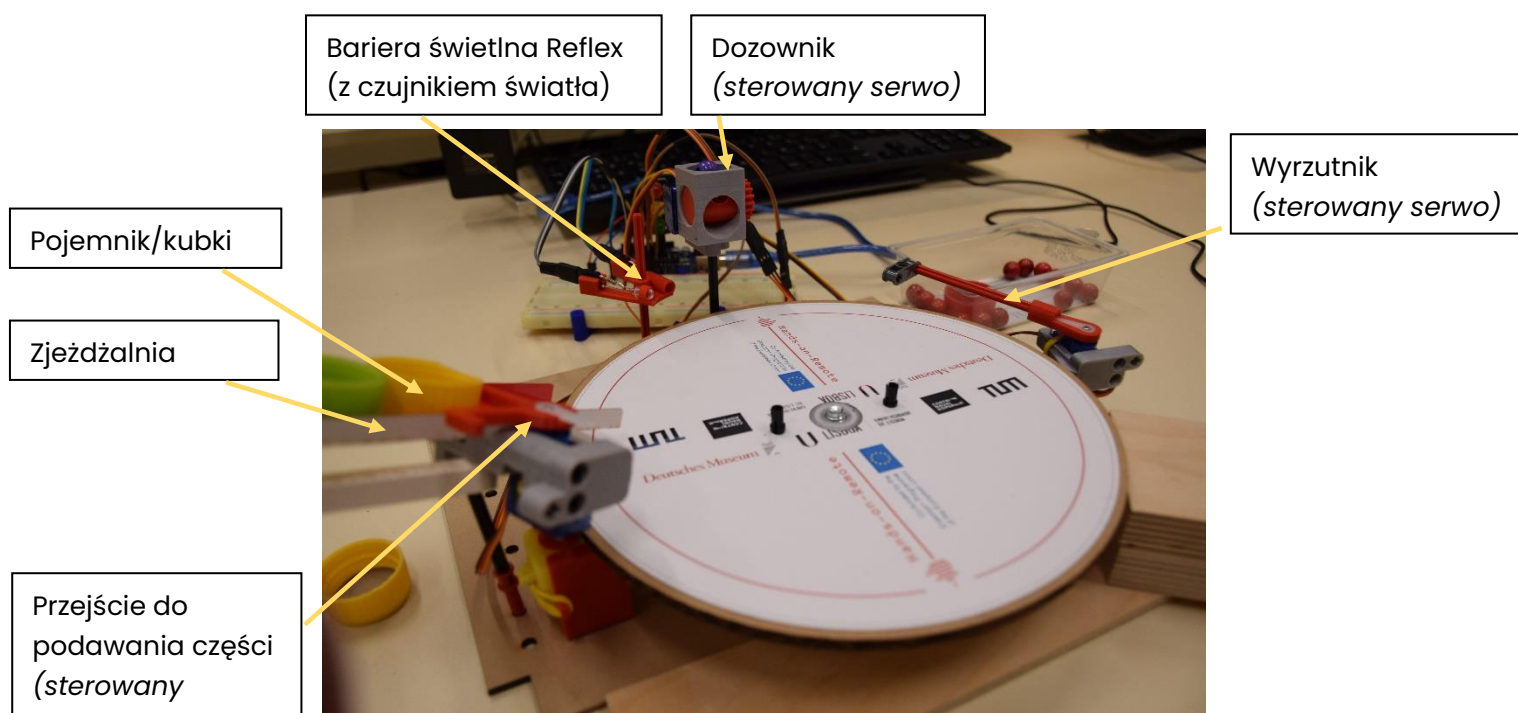
sytuacjach dystansowych i przeciwdziałanie negatywnym skutkom izolacji.

Częścią koncepcji jest wykorzystanie tego podejścia do stworzenia poczucia przynależności, pomimo fizycznej odległości – i po prostu wyzwolenia radości u młodych ludzi.

Jest to bardzo proste podejście, które ma tę zaletę, że jest łatwe do zrozumienia, tanie i szybkie do wdrożenia. Podejście to uwzględnia wcześniejsze doświadczenia nauczycieli i dlatego jest łatwo adaptowalne dla nauczycieli, którzy mają niewielkie doświadczenie z mediami technicznymi w edukacji na odległość.

1.4 Krótkie podsumowanie

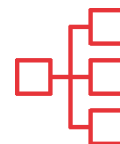
Moduł dydaktyczny "Automatyka w miniaturze" umożliwia uczniom samodzielne zaprogramowanie i uruchomienie małego zakładu produkcyjnego. Moduł sygnalizacji świetlnej jako wskaźnik stanu, bariera świetlna i stół obrotowy służą jako elementy, które wraz z różnymi ruchomymi częściami funkcjonalnymi stają się mini zakładem produkcyjnym.



Uczniowie mogą sterować obrotnicą jako systemem napełniania – bariera świetlna wykrywa pojemnik na obrotnicy, wskaźnik stanu pokazuje zmianę stanu poprzez zmianę koloru z czerwonego na zielony, a dozownik służy do napełniania czegoś do pojemnika.

Proces produkcji jest zależny od czujników – refleksyjna bariera świetlna wykrywa przechodzący pojemnik, a następnie daje impuls do napełnienia kubka.

Struktura jednostek dydaktycznych



Nauczyciele mogą wybrać różne sekwencje i podejścia do lekcji. Lekcje opierają się na sobie nawzajem, ale pozwalają na podział na program wprowadzający i zaawansowany. Kilka motywujących spostrzeżeń na temat przemysłowego zastosowania stolików obrotowych umieszcza klasę na początku w fikcyjnej fabryce. Dla swojej mini-fabryki zespoły mogą wybrać, co "produkują" w swoim własnym fikcyjnym scenariuszu. Jako wprowadzenie do programowania za pomocą Arduino, uczniowie najpierw włączają i wyłączają diody LED wskaźnika stanu. W tym celu pracują zarówno z prawdziwym, jak i wirtualnym modułem sygnalizacji świetlnej. W przypadku pracy z minisystemem produkcyjnym wszystkie zespoły zaczynają razem i np. wprawiają w ruch obrotnicę, dołączają dozownik i programują go, w zależności od sygnałów z czujnika światła. Następnie uczniowie mogą zdecydować, jakie dodatkowe elementy chcą dodać do swojego procesu produkcyjnego.

Programowanie

Poziom trudności można wybrać również na poziomie programowania: Jeśli nauczyciel chce wykorzystać moduł do wprowadzenia, uczniowie piszą tylko kilka linii kodu w podanym w dużej mierze programie. Jeśli nauczyciel chce wykorzystać moduł do zaawansowanego programowania, uczniowie mają dodatkowe zadania, np. definiowanie klas.

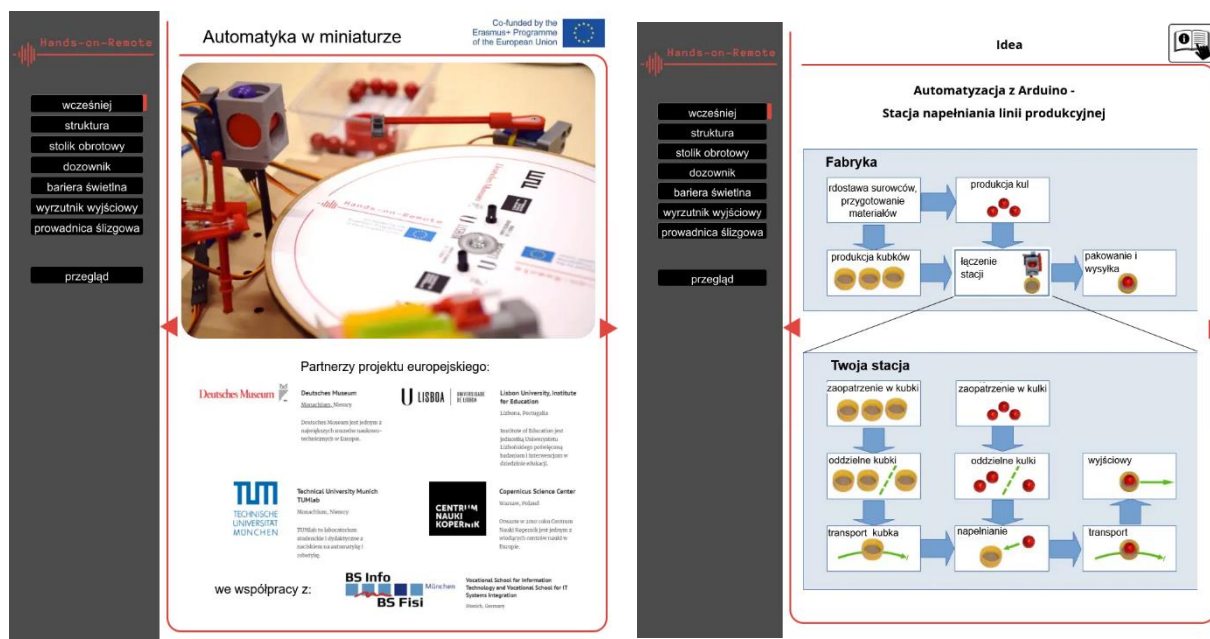
Materiały do modułu dydaktycznego

Poniższe materiały należą do modułu dydaktycznego "Automatyka w miniaturze" i można je pobrać z następujących stron projektu do pobrania

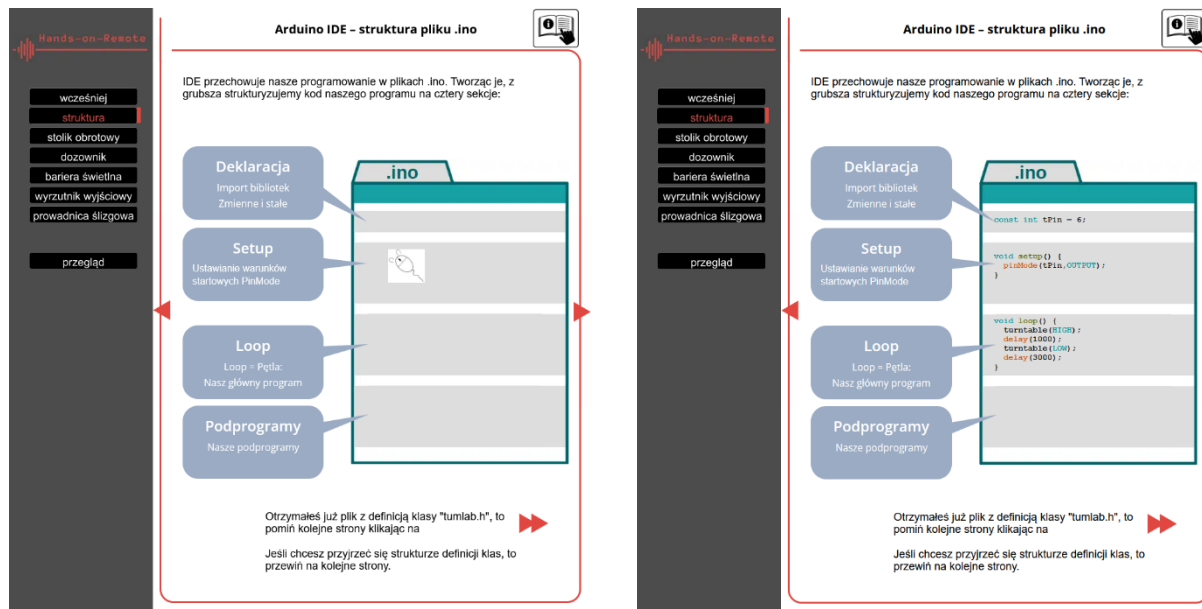
<https://sites.google.com/campus.ul.pt/hands-on-remote-de/downloads>

Página do projecto do Museu Alemão: <https://www.deutsches-museum.de/forschung/forschungsinstitut/projekte/detailseite/erasmus-hands-on-remote>

- Plik z prezentacją dla nauczyciela, *Plik z prezentacją dla nauczyciela (pptx)*, jak również film na temat obwodu, *Video_Circuit (mp4)*
- Interaktywna multimedialna strona instruktażowa dla uczniów, instrukcja krok po kroku do samodzielnej pracy z miniprodukcją, *interaktywna strona instruktażowa dla uczniów PL (html)*



- Arkusze ćwiczeniowe do jednostki "Automatyka w kontekście społecznym" oraz link do tablicy online, *Arkusze ćwiczeniowe Automatyka w kontekście społecznym (pdf)*
- Pliki do druku 3D i wycinania laserowego dla zestawu materiałów, *zestaw materiałów (lista, pliki stl i wycinane laserowo)*
- Plik z typowymi błędami i trudnościami, *Troubeshooting_Automation_in_Miniature (pdf)*.



Poszczególne kroki dotyczące produkcji zestawów materiałowych, jak również budowy i programowania całego systemu wykraczają poza zakres niniejszego opisu modułu dydaktycznego i znajdują się w wymienionych wyżej plikach towarzyszących. W przypadku mini zakładu produkcyjnego budowa i programowanie są przedstawione w szczególności na stronie instrukcji multimedialnej. Zrzuty ekranu w dodatku zapewniają wstępny wgląd w strukturę i wygląd strony instrukcji (patrz rozdział 6, Dodatek: Wgląd w multimedialną stronę instrukcji). Daje to lepsze wyobrażenie o tym, jak zdalnie pracować z Arduino i systemem.

2 Powiązanie z programem nauczania

Konstrukcja modułu, zwłaszcza zakładu produkcyjnego mini-P i materiałów towarzyszących, została oparta na analizach potrzeb z udziałem nauczycieli z różnych szkół zawodowych. Moduł ma powiązania z programami nauczania różnych zawodów szkoleniowych oraz z lekcjami w technikum. Zawody, dla których moduł może odegrać rolę, to np. informatyk, technik elektronik, technik automatyk, elektryk przemysłowy, programista. Jednostka dydaktyczna dotycząca społecznych aspektów automatyzacji związana jest z etyką stosowaną (etyką techniki). Słowa kluczowe dla tematów to np. systemy i procesy, złożone systemy techniczne, technika sterowania i programowanie obiektowe. Powiązania z programem nauczania istnieją generalnie dla klas 10–13.

3 Przegląd sekwencji modułów dla modułu "Automatyka w miniatursze".

Jednostka 1.a Wprowadzenie do zastosowania obrotnic w automatyce oraz cel miniaturyzacji procesu automatyzacji. (15 min)		Jednostka 1.b Wprowadzenie do pracy z Arduino z wykorzystaniem pierwszego projektu obwodu i przykładu programowania z różnokolorowymi diodami LED. (30–45 min)		
Jednostka 2.a Wprowadzenie środowiska symulacyjnego online do programowania Arduino w celu umożliwienia pracy nad programowaniem w zespole w sytuacji dystansowej. Zaprojektowanie symulowanego obwodu dla serwomotoru i zaprogramowanie serwomotoru. (15 min)		Jednostka 2.b Wprowadzenie do serwomotorów, w tym obwody i programowanie rzeczywistego serwomotoru. (30 min)		
Jednostka 3.a Budowa stolika obrotowego, obwody, programowanie motoreduktora. (20 min)		Jednostka 3.b Zastosowanie fotokomórki, w tym obwód, programowanie i wyświetlanie wartości na ploterze. (25 min)		
Unit 4 Uczniowie pracują samodzielnie i wybierają elementy, które chcą dodać do procesu automatyzacji. (min. 45 min)				
Wersja	Wariant	Wariant	Wariant	Wersja długa
Hub Bariera świetlna Dozownik	Hub Bariera świetlna Dozownik Wskaźnik stanu LED dla urządzenia	Hub Bariera świetlna Dozownik Wyrzutnik	Hub Bariera świetlna Dozownik Wyrzutnik Zjeżdżalnia wejściowa	Hub Bariera świetlna Dozownik Wyrzutnik Zjeżdżalnia wejściowa Wskaźnik stanu LED dla napędów ustawczych
Unit 5 Współpraca uczniów: łączenie w pary piastów różnych zespołów. (min. 45 min)				
Jednostka 6 Współpraca cyfrowa: Wykorzystanie transmisji światła do wywołania efektów fizycznych w odległych miejscach. (elastyczny czas trwania)				
Unit 7 Automatyka w kontekście społecznym. (min. 45 min)				
Jednostka 8 Inne zastosowania stolika obrotowego (patrz tabela poniżej) (min. 45 min)				
Maszyna sortująca		Odtwarzacz płyt świetlnych		
Hub Bariera świetlna Wskaźnik stanu LED Jasność/kolor Wyrzutnik do sortowania		Stolik z papierowymi nakładkami z czarno-białymi wzorami lub szarościami Czujnik światła jako głowica pomiarowa Głośnik (nie wchodzi w skład zestawu materiałowego)		

4 Treść: Kolejność uczenia się dla modułu nauczania "Automatyka w miniaturze".

4.1 Jednostka 1.a: Wprowadzenie do tematu

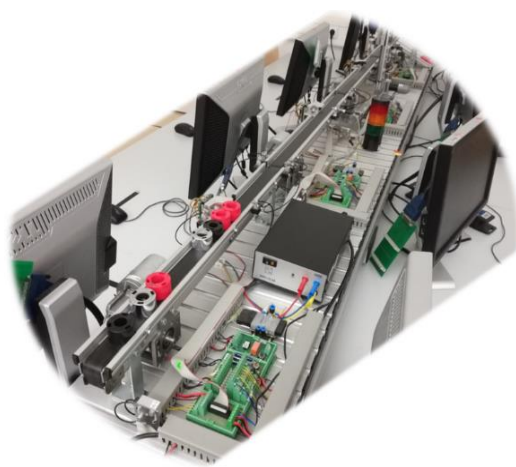


Temat: Różne praktyczne przykłady procesów automatyzacji

Cel: Wprowadzenie do tematu automatyzacji i celu miniaturyzacji procesu automatyzacji.

Czas trwania: ok. 15 min.

Tło/wprowadzenie do pomysłu:



Wielu uczniów szkół zawodowych będzie miało w swoim życiu zawodowym do czynienia z procesami automatyzacji – albo dlatego, że w przyszłości sami będą programować systemy produkcyjne, albo dlatego, że będą pracować z takimi systemami – na przykład

w technologii żywności.

Ogólnie rzecz biorąc, temat automatyzacji oferuje wiele praktycznych odniesień ze świata rzeczywistego.

Idea mini-automatyzacji sięga większej linii produkcyjnej w TUMlab. W tym eksperymentalnym laboratorium odpowiedni "proces produkcyjny" jest programowany i automatyzowany przez studentów. Nauczyciele wielokrotnie wyrażali życzenie, czy nie ma możliwości stworzenia linii produkcyjnej w mniejszym formacie, którą młodzi ludzie mogliby programować w szkole – a teraz także w domu. W ten sposób powstał pomysł wykorzystania Arduino do umożliwienia procesu automatyzacji na małą skalę.

Format: online poprzez wideokonferencję lub w klasie



Materiały dla nauczyciela: prezentacja PowerPoint, w tym linki do filmów z YouTube.

Materiały dla uczniów: Brak

Treść:

Działania nauczycieli

Wykorzystując fragmenty różnych filmów z YouTube, nauczyciele zapewniają wgląd w rzeczywiste praktyczne przykłady zautomatyzowanych zakładów przemysłowych. Filmy pokazują ważną rolę, jaką odgrywają obrotnice w procesach automatyzacji. Nauczyciele przedstawiają pomysł wykorzystania obrotnicy zamiast przenośnika taśmowego w zminiaturyzowanym procesie automatyzacji. Obrotnica może być stosowana w różnych formach:

- albo jako tarcza transportowa systemu automatyzacji z kilkoma etapami pracy
- lub jako pojedynczy stół obrotowy, np. jako pojedyncza stacja obróbki kompletnej linii produkcyjnej, jak np. stacja napętniania.

Ponadto jeden z filmów pokazuje samodzielnie zbudowaną maszynę, która wykorzystuje zautomatyzowany proces do umieszczania pizzy na stole obrotowym. Uczniowie mogą wykorzystać ten pomysł do nawiązania do własnego życia codziennego.



Działania uczniów

Młodzi ludzie oglądają filmy.

4.2 Jednostka 1.b: Wprowadzenie do podstawowych obwodów i programowania Arduino

Tematy: Projektowanie obwodów, podstawowe polecenia programistyczne (Arduino IDE), częste błędy w programowaniu.

Cele: Studenci po raz pierwszy aktywnie poznają mikrokontroler. Uczą się podstawowych metod pracy przy projektowaniu układu i programowaniu mikrokontrolera. Na początek młodzież pracuje z diodami LED lub modułem sygnalizacji świetlnej, który później może służyć jako wskaźnik stanu dla procesów automatyzacji.

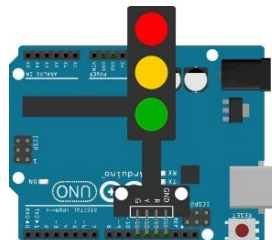
Tło/wprowadzenie do pomysłu: Mikrokontrolery są wbudowane w wiele codziennych urządzeń technicznych, w których jednak często pozostają niewidoczne. Mikrokontroler potrzebuje do swoich funkcji zarówno obwodów, jak i programowania, co daje nauczycielom możliwość wprowadzenia obu tematów. Dla uczniów motywujące jest to, że dzięki swojej pracy mogą bezpośrednio uzyskać widoczny wynik. Jeśli młodzi ludzie nie osiągną pożądanego rezultatu, sami lub ich nauczyciele mogą szybko zidentyfikować i poprawić błędy.

Czas trwania: ok. 30 min, z wykorzystaniem filmu o obwodach i wypróbowaniem wirtualnego laboratorium DC, ok. 45 min.



Materiały dla nauczyciela: prezentacja PowerPoint

Materiały dla uczniów: Arduino, Arduino IDE, hub USB, diody LED lub moduł sygnalizacji świetlnej. Jeśli nauczyciel chce z nich skorzystać: film o obwodach, przenośna wersja Arduino IDE oraz pliki na pamięci USB.



Ważna uwaga: Arduino jest podłączone do komputera poprzez hub USB. Za każdym razem, gdy budowany jest obwód, Arduino jest odłączane od huba USB. Procedura ta służy ochronie komputera w przypadku ewentualnego zwarcia.

Format: online poprzez wideokonferencję lub w sali komputerowej w szkole

Treść:

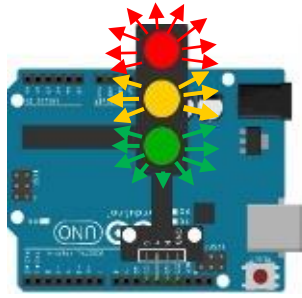
Działania nauczycieli

- Nauczyciele zapoznają uczniów z Arduino i podstawowymi krokami projektów Arduino.
- Nauczyciele wprowadzają tekstowe środowisko programistyczne Arduino IDE.
- *W zależności od poziomu wiedzy uczniów:*
 - Nauczyciele przypominają młodzieży prosty obwód, za pomocą którego zapalają się diody LED.
 - Nauczyciele prowadzą uczniów w budowaniu obwodu na Arduino dla diod LED lub modułu sygnalizacji świetlnej.
 - Nauczyciele pytają młodzież o postępy i proszą o pokazanie migających diod LED.
 - Nauczyciele pomagają uczniom w rozwiązywaniu problemów technicznych, korzystając z pliku zawierającego listę typowych błędów i trudności.
 - W nauczaniu na odległość nauczyciele pozwalają młodym ludziom dzielić się ekranem w wideokonferencji, aby pomóc w problemach z programowaniem.
 - Nauczyciele łączą programowanie modułu sygnalizacji świetlnej z automatyką, wyjaśniając uczniom zastosowanie diod LED lub świateł drogowych jako wskaźników stanu w systemach automatyki.



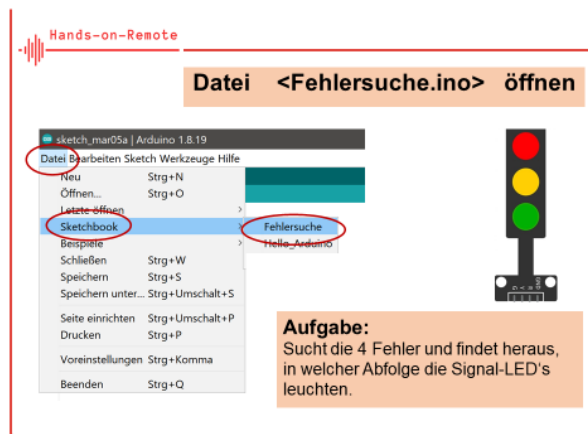
Działania uczniów

- Młodzież ogląda film o obwodach elektrycznych i buduje swój własny obwód elektryczny online za pomocą DC Virtual Lab (https://phet.colorado.edu/sims/html/circuit-construction-kit-dc-virtual-lab/latest/circuit-construction-kit-dc-virtual-lab_de.html).
- Uczniowie budują swój pierwszy obwód z diodą LED lub modulem sygnalizacji świetlnej na prawdziwym Arduino.
- Młodzi ludzie podłączają Arduino i wgrywają pierwszy kod, dzięki któremu powstaje lampa błyskowa LED.
- Pierwszym zadaniem programistycznym dla uczniów jest zmiana kodu programu w celu zmiany czasu migania diody LED.
- Młodzi ludzie radzą sobie z problemami technicznymi, korzystając z pliku zawierającego listę typowych błędów i trudności. - W razie potrzeby mogą omówić swoje programowanie z nauczycielem.



- Zadaniem uczniów jest teraz sprawienie, aby 3 diody LED migaly jednocześnie. W tym momencie nauczyciel może przejść na programowanie obiektowe: 3 diody LED = kilka obiektów tego samego typu. W programowaniu obiektowym diody LED mogą być zatem pogrupowane w klasę.

Studenci, którzy posiadają już pogłębioną wiedzę na temat programowania obiektowego, mogą definiować własne klasy. Wszyscy inni mogą korzystać z predefiniowanych definicji klas. Użycie predefiniowanych definicji klas upraszcza programowanie w całym procesie.



- Studenci pogłębiają znajomość języka programowania poprzez wyszukiwanie typowych błędów semantycznych/syntaktycznych w gotowym pliku do debugowania.

4.3 Jednostka 2.a: Symulacja online

Nauczyciele mogą kontynuować w tej kolejności lub wprowadzić najpierw obrotnicę (3.a) i barierę świetlną (3.b).

Tematy: Poznanie i wykorzystanie internetowego środowiska symulacyjnego, które umożliwia identyfikację typowych błędów programistycznych i wymianę programów między członkami zespołu, wstępne programowanie serwomotoru.

Cele: Uczniowie poznają i wykorzystują środowisko symulacyjne online "Wokwi". Może to być szczególnie ważne, jeśli chcą wymieniać się programami w zespole w sytuacji odległości. Przekazują kody pomiędzy Wokwi a Arduino IDE.

Tło/wprowadzenie do pomysłu: Ze strony nauczycieli pojawiła się chęć wykorzystania tych samych metod w nauczaniu bezpośrednim i online. Nauczyciele mogą wykorzystywać symulację online Wokwi zarówno na miejscu, jak i w nauczaniu na odległość. Uczniowie mogą ją wykorzystać do utrwalenia zdobytej wiedzy na temat rozwiązywania problemów w środowisku programistycznym. Ponieważ efekty są bezpośrednio widoczne w symulacji online, czynnik motywujący, jakim jest zobaczenie natychmiastowego rezultatu własnej pracy, może być zachowany dla młodych ludzi również w środowisku online.

Przejdzie z symulacji z plikiem debugowania do modułu sygnalizacji świetlnej z tymi samymi błędami umożliwia uczniom rozpoznanie prawdziwych obiektów w symulacji. Ponadto młodzież może dodać nowy element do swojego kodu, a później przenieść tę wiedzę na prawdziwe Arduino. Ponadto symulacja online daje duże korzyści praktyczne przy dzieleniu się kodami.

Czas trwania: ok. 15 min.



Materiały dla nauczyciela: prezentacja PowerPoint

Materiały dla uczniów:

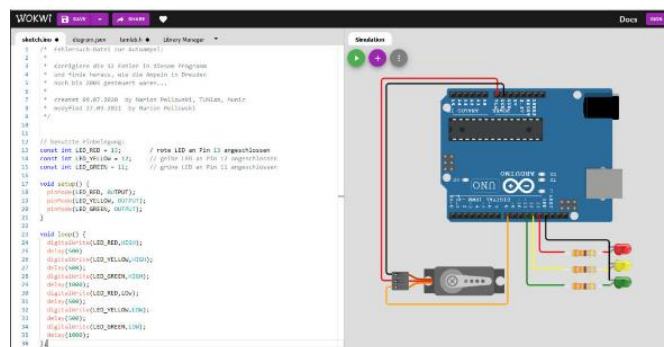
Gotowy plik Wokwi do rozwiązywania problemów (<https://wokwi.com/projects/328915985532715602>), połączenie z Internetem

Format: Uczniowie pracują samodzielnie z plikiem Wokwi, albo online i ze wsparciem nauczyciela poprzez wideokonferencję, albo w sali komputerowej w szkole.

Treść:

Działania nauczycieli

- Rozdanie pliku Wokwi i wyjaśnienie zadania



Działania uczniów

- Młodzi ludzie znajdują błędy w pliku rozwiązywania problemów Wokwi i mogą obserwować efekty swoich zmian na symulowanych diodach LED.
- Uczniowie dodają do symulacji serwowymotor.

4.4 Dział 2.b: Serwowymotor

Tematy: Poznanie serwowymotora i jego funkcji, programowanie pozycji kątowych serwowymotora.

Cele: uczniowie poznają właściwości i funkcje serwowymotorów. W zminiaturyzowanym zakładzie produkcyjnym zespoły uczniowskie wykorzystują serwowymotory jako siłowniki, np. do poruszania kół zębatych lub szlabanu na przejazd. Ogólnie rzecz biorąc, zastosowanie serwowymotorów jako siłowników jest powszechne w mini automatyce lub modelarstwie.

Tło/wprowadzenie do pomysłu: Trzy serwowymotory poruszają zasadnicze elementy mini linii produkcyjnej: jeden serwowymotor tworzy przejście dla zsypu wejściowego, jeden serwowymotor umożliwia wydawanie kulek w dozowniku, a jeden serwowymotor porusza wyrzutnikiem. Uczniowie mogą więc przenieść zdobytą wcześniej wiedzę z zakresu programowania serwowymotorów do różnych zastosowań.

Jeśli nauczyciel chce to wprowadzić, uczniowie mogą dowiedzieć się, że różne obiekty rzeczywiste tego samego typu w programowaniu są różnymi obiektami tej samej klasy - ważny wgląd w programowanie obiektowe.

Czas trwania: ok. 30 minut



Materiały dla nauczyciela: prezentacja PowerPoint

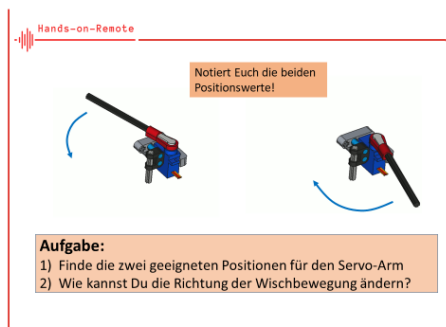
Materiały dla uczniów: Link do symulacji Wokwi (<https://wokwi.com/projects/328915985532715602>), multimedialny plik HTML, Arduino, Arduino IDE, serwomechanizmy, breadboard, kable zworkowe. Jeśli nauczyciel chce z nich skorzystać: przenośna wersja Arduino IDE na pamięci USB.

Format: online lub w sali komputerowej w szkole

Treść:

Działania nauczycieli

– Wstęp: Co może zrobić serwowmotor? Do czego chcemy go wykorzystać?



Działania uczniów

– Młodzi ludzie mogą wybrać, czy chcą najpierw pracować z dozownikami, barierą dla przejścia czy wyrzutnikiem. Serwowmotor przełącza się tylko między dwoma pozycjami; przełącza się między pozycją spoczynkową a pozycją aktywną.

– Uczniowie mogą skopiować wymagany nowy kod z symulacji do Arduino IDE lub odwrotnie.

– Uczniowie budują obwód z prawdziwym serwowmotorem i Arduino i próbują znaleźć odpowiednie pozycje spoczynkowe lub aktywne.

4.5 Jednostka 3.a: Obrotnica

Tematy: Zastosowanie praktycznych umiejętności w składaniu części mechanicznych, tworzenie kolejnego prostego obwodu oraz wstawianie nowego kodu programistycznego w celu stworzenia nowego obiektu.

Cel: Uczniowie zdobywają praktyczne doświadczenie przy składaniu stolika obrotowego i montowaniu silnika napędowego. Mogą być zmuszeni do rozwiązywania małych problemów mechanicznych podczas montażu, np. do znalezienia właściwej pozycji dla silnika napędowego. Budują kolejny obwód z Arduino i dodają nowy kod, aby poruszyć nowy element.

Tło/wprowadzenie do pomysłu: Obrotnica jest centralnym elementem zminiaturyzowanego procesu automatyzacji, ponieważ umożliwia transport obiektów pomiędzy poszczególnymi stacjami procesu automatyzacji.

Format: Uczniowie pracują samodzielnie ze wskazówkami zawartymi w multimedialnym pliku HTML, albo online i z pomocą nauczyciela poprzez wideokonferencję, albo w laboratorium komputerowym w szkole.

Czas trwania: ok. 20 min.



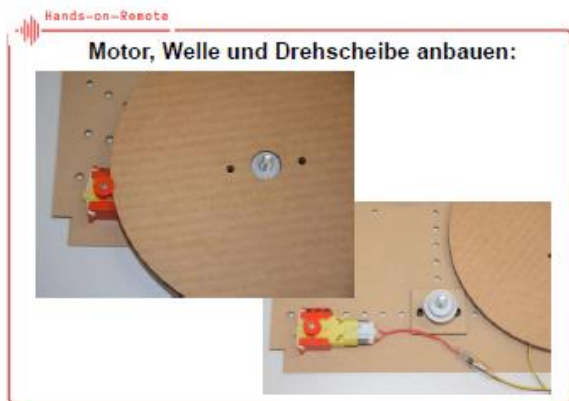
Materiały dla nauczyciela: prezentacja PowerPoint, w tym linki do filmów z YouTube.

Materiały dla uczniów: Multimedialny plik HTML, Arduino IDE, płyta bazowa, stolik obrotowy, motoreduktor z kołem ciernym, mocowanie silnika, gumowy pierścień, śruba z dwiema nakrętkami jako oś, czarne kołki Lego, Arduino, płytka z wtyczkami, kabel ze zworkami.

Treść:

Działania nauczycieli

- Krótkie wprowadzenie do budowy stolika z wykorzystaniem prezentacji PowerPoint



Działania uczniów

- Młodzież montuje stolik i silnik napędowy na płycie bazowej.
- Uczniowie rozbudowują obwód Arduino i podłączają silnik napędowy.

```
✓ → 📄 ⬆ ⬇  
Kugelfabrik § tumlab.h  
  
//Create a motor object  
  
Motor xxx(6);  
...  
xxx.go(HIGH);  
...  
xxx.go(LOW);
```

- Uczniowie dodają nowe linie kodu dla silnika napędowego. Tworzą obiekt silnika i włączają lub wyłączają silnik. Kod potrzebny do włączania i wyłączania silnika jest bardzo podobny do kodu potrzebnego do tego, aby dioda LED mrugała, więc uczniowie mogą przenieść wcześniej zdobytą wiedzę. Ich zadanie polega na przesunięciu stolika o ćwierć obrotu.

4.6 Jednostka 3.b: Bariera świetlna

Tematy: Budowa obwodu dla refleksyjnej bariery świetlnej, rezystor zależny od światła (LDR) jako czujnik jasności, zależność wartości rezystancji od natężenia światła, sterowanie procesami zależne od czujnika.

Cel: Uczniowie uświadamiają sobie, że czujniki są niezbędne do automatycznego sterowania zakładem produkcyjnym. Dowiadują się jak działa fotokomórka refleksyjna i wykorzystują wbudowany czujnik światła do sterowania swoim "procesem produkcyjnym".

Czas trwania: ok. 25 min.



Materiały dla nauczyciela: prezentacja PowerPoint, w tym link do filmu na YouTube

Materiały dla uczniów: Multimedialny plik HTML, Arduino IDE, zmontowany już stolik, LDR + LED z uchwytem jako bariera światła refleksyjnego, zmontowany już układ na Arduino i płytce wtyczkowej, więcej kabli zworkowych.

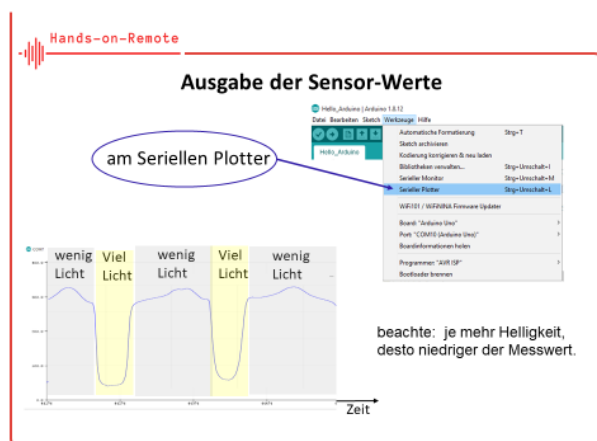
Tło/wprowadzenie do pomysłu: Bariery świetlne są często używanymi elementami w życiu codziennym, np. przy zamykaniu drzwi S-Bahn. Dzięki mini zakładowi produkcyjnemu młodzi ludzie poznają możliwe zastosowanie w procesach automatyzacji. Dowiadują się, jak działa opór zależny od światła i jak na zmierzone wartości mogą wpływać nawet niewielkie odchylenia jasności. Nauczyciel może wykorzystać te pomiary, aby pokazać uczniom trudności związane z dokładnym i znaczącym pomiarem i przedyskutować z zespołami, co to oznacza dla procesów zależnych od czujników na przykładzie ich mini zakładu produkcyjnego.

Format: Uczniowie pracują samodzielnie ze wskazówkami zawartymi w multimedialnym pliku HTML, albo online i z pomocą nauczyciela poprzez wideokonferencję, albo w laboratorium komputerowym w szkole. W środowisku online, uczniowie mogą pracować razem w pokojach przerw.

Treść:

Działania nauczycieli

- Nauczyciel przedstawia typowe zastosowanie refleksyjnych barier światlnych w procesach automatyzacji.
- Nauczyciel wprowadza rezystor zależny od światła (LDR) jako czujnik jasności.



- Nauczyciel wspiera uczniów w posługiwaniu się fotokomórką i interpretacji krzywych wartości mierzonych.
- Nauczyciele są dostępni, by odpowiadać na pytania uczniów, np. za pośrednictwem videokonferencji.



Działania uczniów

- Młodzież poznaje właściwości refleksyjnej bariery światłnej korzystając z instrukcji multimedialnej.
- Uczniowie montują fotokomórkę refleksyjną na płycie bazowej.
- Młodzież rozbudowuje układ Arduino i podłącza do niego LDR/odblaskową barierę światlną.
- Uczniowie wyświetlają wartości czujników uzyskane z LDR za pomocą plotera Arduino i interpretują wykresy. W ten sposób mogą zrozumieć zmiany wartości w różnych warunkach oświetleniowych.
- Uczniowie za pomocą refleksyjnej bariery światłnej wykrywają, czy przed czujnikiem światła znajduje się filiżanka. W zależności od tego, czy tak jest, mogą zatrzymać lub uruchomić stół obrotowy.
- Uczniowie dodają nowe linie kodu dla bariery światłnej, w szczególności polecenie `waitForCan` – dopiero gdy czujnik wykryje kubek, program będzie kontynuowany.

4.7 Jednostka 4: Rozszerzenia procesu automatyzacji

Tematy: Transfer wcześniej zdobytej wiedzy na nowe elementy

Cel: Uczniowie pracują samodzielnie i mogą wybrać, które elementy chcą włączyć do procesu automatyzacji. W ten sposób mogą przenieść zdobytą już wiedzę o programowaniu, obwodach lub procesach automatyzacji i zastosować ją do nowych zadań. Daje im to również swobodę, której często pragną, aby przeprowadzać własne eksperymenty.

Czas trwania: elastyczny, min. 45 minut



Materiały dla nauczyciela: w razie potrzeby: plik z wykazem typowych błędów i trudności.

Materiały dla uczniów: Multimedialny plik HTML, Arduino IDE, zmontowany już stolik, LDR + LED z uchwytem jako bariera świetlna refleksyjna, Arduino, płytka z wtyczkami, kable zworkowe, jako materiał dla uczniów: wewnątrz do wyboru: Dozownik, wyrzutnik, zsyg wejściowy, diody LED lub moduł sygnalizacji świetlnej.

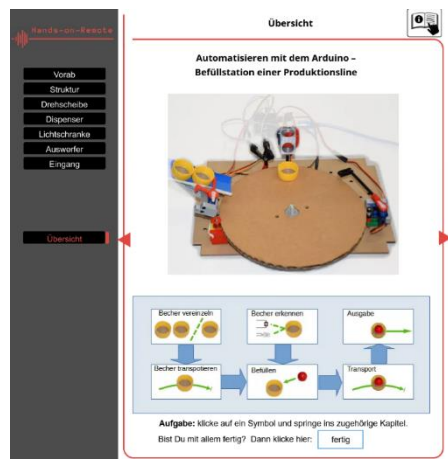
Tło/wprowadzenie do pomysłu: Uczniowie mogą zbudować swój proces automatyzacji z kilku elementów funkcjonalnych.

Format: Uczniowie pracują samodzielnie ze wskazówkami zawartymi w multimedialnym pliku HTML, albo online i z pomocą nauczyciela poprzez wideokonferencję, albo w laboratorium komputerowym w szkole. W środowisku online, uczniowie mogą pracować razem w pokojach przerw.

Treść

Działania nauczycieli

Nauczyciele są dostępni w przypadku pytań ze strony uczniów, np. za pośrednictwem wideokonferencji.



Działania uczniów

Młodzież pracuje z multimedialnym plikiem HTML, aby dodać kolejne elementy do swojego procesu automatyzacji. Na stronie przeglądowej multimedialnego pliku HTML uczniowie mogą kliknąć na różne elementy, które ich interesują. Młodzież może wybrać elementy, które chce dodać do procesu automatyzacji.

Wersja	Wariant	Wariant	Wariant	Wersja długa
Hub Bariera światlna Dozownik	Hub Bariera światlna Dozownik Wskaźnik stanu LED dla dozownika	Hub Bariera światlna Dozownik Wyrzutnik	Hub Bariera światlna Dozownik Wyrzutnik Zjeżdżalnia wejściowa	Hub Bariera światlna Dozownik Wyrzutnik Zjeżdżalnia wejściowa Wskaźnik stanu LED

Jednostka ta może być realizowana elastycznie w różnych wariantach (patrz tabela powyżej), w zależności od wymagań czasowych i merytorycznych nauczyciela. Przy różnych wariantach nauczyciele mogą również uwzględnić różne umiejętności programowania i wcześniejsze doświadczenia uczniów.

4.8 Jednostka 5: Współpraca uczniów



Tematy: Poszerzanie umiejętności programowania, praca w zespole i współpraca

Cel: Wspólna praca uczniów

Czas trwania: min. 45 min.



Materiały dla nauczyciela: Brak

Materiały dla uczniów: Mini-zakłady produkcyjne już zbudowane w zespołach

Niezbędna wiedza wstępna: Zespoły studenckie muszą mieć zaprogramowaną przynajmniej jedną z części funkcjonalnych procesu automatyzacji. Najlepiej, jeśli studenci w różnych zespołach zaprogramowali różne części funkcjonalne.

Format: W pracowni komputerowej w szkole

Treść

Działania nauczycieli

Nauczyciel inicjuje i wspiera współpracę i działania zespołowe uczniów.



Działania uczniów

Zadaniem młodzieży jest połączenie stolików obrotowych w poprzek zespołów. Uczniowie decydują w jaki sposób chcą sparować stoliki i wspólnie programują niezbędne czynności programistyczne.



Tematy: Pomiary za pomocą czujników, zależne od czujników wyzwalanie różnych elementów wykonawczych procesu automatyzacji, także w odległych miejscach.

Cel: Promowanie poczucia wspólnoty wśród uczniów.

Czas trwania: Elastyczny



Materiały dla nauczyciela: Brak

Materiały dla uczniów: Bariera refleksyjna, dioda LED

Niezbędna wiedza wstępna: Studenci powinni wcześniej zaprogramować proces zależny od czujnika, w tym zależne od czujnika wyzwalanie aktuatora (jednostka 3.b).

Format: online lub w pracowni komputerowej w szkole

Treść:

Działania nauczycieli

Nauczyciele zapoznają uczniów z ideą procesu komunikacji poprzez transmisję światła i wspierają młodzież w ustawianiu czujników.



Działania uczniów

Młodzi ludzie programują w automatyce siłownik tak, aby poruszał się, gdy czujnik światła wykryje wzrost jasności. Członkowie zespołu uczestniczą razem w wideokonferencji. Pierwszy uczeń ustawia diodę LED przed kamerą internetową do wideokonferencji. Drugi uczeń ustawia refleksyjną barierę świetlną przed własnym widokiem wideokonferencyjnym tak, że czujnik światła wykrywa sygnał świetlny diody LED pierwszego ucznia. Gdy pierwszy uczeń włączy światło diody LED, czujnik światła rozpoznaje wzrost jasności u drugiego ucznia i wprawia tam w ruch siłownik. Partner zespołu może teraz przekazać sygnał świetlny do trzeciego ucznia itd.

4.10 Jednostka 7: Automatyka w kontekście społecznym

Nauczyciele mogą kontynuować w tej kolejności lub wprowadzić najpierw inne zastosowania wężła (Unit 8).

Tematy: Zachowania konsumentów, uzależnienie od produkowanych dóbr, skutki uboczne konsumpcji (zapotrzebowanie na energię, odpady, transport itp.), zalety i wady procesów automatyzacji.

Cele: uczniowie badają różne skutki, jakie procesy automatyzacji mogą mieć w społeczeństwie. Młodzież zastanawia się nad własnymi zachowaniami konsumenckimi.

Czas trwania: elastyczny, min. 45 min.



Materiały dla nauczyciela: dostęp do tablicy online – dostęp rozdaje uczniom nauczyciel. Karty z pytaniami, które nauczyciel może rozdać uczniom.

Materiały dla uczniów: Karty z pytaniami, tablica internetowa i łącze internetowe, jeśli nauczyciel chce z nich korzystać: już ustawione stoliki

Format: online lub w sali komputerowej w szkole

Treść:

Działania nauczycieli

Nauczyciel rozdaje uczniom karty z pytaniami i/lub prowadzi dyskusję w grupie.

Nauczyciel wprowadza i przechodzi do nowego tematu, gratulując uczniom osiągnięć i pytając ich, jak chcieliby się poczęstować po wykonaniu zadań. Jeśli nauczyciel może wdrożyć odpowiedź bezpośrednio, to zaprasza do włączenia jej do swojej lekcji.



Herzlichen Glückwunsch zu Eurer ersten automatisierten Fertigung!

Jetzt habt Ihr richtig was geschafft, jetzt könnt Ihr Euch etwas Schönes gönnen...

Was würdet Ihr Euch denn im Augenblick gern gönnen?

Diskutiert gern im Team, was Euch so einfällt und wenn Ihr mögt, notiert es hier...



Gibt es bei Euren Ideen, was Ihr Euch jetzt gern gönnen würdet, etwas, das ohne Konsum und ohne Produkte auskommt?

Könnt Ihr Euch vorstellen, Euch etwas zu gönnen, ohne dafür etwas zu brauchen oder verbrauchen? Was wäre das?

Nauczyciel omawia odpowiedzi z młodzieżą i rozróżnia, czy konsumpcja produktów jest jedną z proponowanych nagród, czy nie. Jeśli wszystkie odpowiedzi opierają się na konsumpcji, nauczyciel rozdaje drugą kartę z pytaniem, czy możliwa byłaby również nagroda bez konsumpcji. Nauczyciel omawia z uczniami skutki uboczne wysokiej konsumpcji (koszty energii, odpady, transport itp.).

W zależności od przebiegu dyskusji i wybranego tematu, nauczyciel zadaje kolejne pytania. Nauczyciel zapoznaje uczniów z tablicą online i wyjaśnia ich zadanie. Nauczyciel może pozwolić młodzieży wybrać temat, który chciałaby dokładniej omówić. Jeśli nauczyciel chce to wypróbować, może użyć papierowego szablonu z sugerowanymi tematami dla centrum dyskusyjnego i pozwolić uczniom wybrać w ten sposób swój temat.





Działania uczniów

- Młodzież dyskutuje o tym, czym chciałaby się poczęstować i wspólnie z nauczycielem zastanawia się nad swoimi zachowaniami konsumpcyjnymi.
- Patrząc na swój samodzielnie zbudowany mini-zakład produkcyjny, uczniowie mogą zauważyć, że on sam również jest czymś "wyprodukowanym" – które części zakładu pochodzą z procesu automatyzacji, które są samodzielnie wykonane?
- Na tej podstawie uczniowie mogą omówić zalety i wady procesów automatyzacji. Mogą również zastanowić się nad zaletami i wadami materiałów produkowanych samodzielnie. Młodzież może zebrać swoje spontaniczne pomysły i myśli, które kojarzą im się z tematem automatyzacji i jej społecznych konsekwencji na tablicy online. Mogą dodać do tablicy kolejne materiały i linki.
- Młodzi ludzie mogą teraz wybrać konkretny temat, który chcieliby pogłębić i omówić go w grupach.

4.11 Jednostka 8: Inne zastosowania stolika obrotowego

Tematy: Użyj wartości jasności, aby wyróżnić obiekty lub odtworzyć dźwięki.

Cel: Zarówno młodzież jak i nauczyciele przynieśli pomysł innego zastosowania materiałów, aby kontynuować pracę z tym samym zestawem materiałów i spróbować czegoś nowego. Niektórzy z nich byli bardzo zmotywowani do dalszej pracy z zestawem materiałów w wolnym czasie. Uczniowie mogą kontynuować pracę z dodatkowymi aplikacjami we własnym zakresie.

Jednocześnie te inne aplikacje mogą być alternatywą, jeśli nauczyciele i uczniowie nie chcą zbyt głęboko zagłębiać się w programowanie.

Czas trwania: elastyczny, min. 45 min



Materiały dla nauczyciela: W razie potrzeby: Drukarka do wydrukowania szablonów papieru stworzonych przez uczniów z różnymi odcieniami szarości

Materiały dla uczniów: Arduino, płyta z wtyczkami, przewody ze zworkami, Arduino IDE, zmontowany już stolik, bariera świetlna

Dodatkowo dla sortownika: wyrzutnik i wskaźnik stanu LED

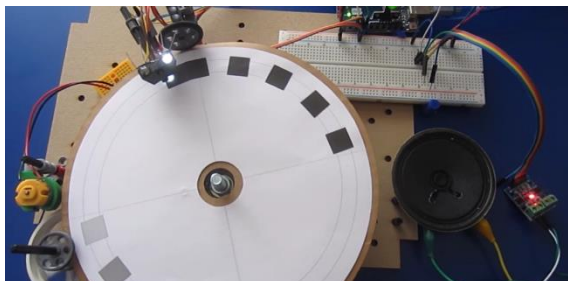
Dodatkowo dla odtwarzacza płyt świetlnych: papier na kartonie wielkości płyty, aby stabilnie umieścić wzór jasności na stoliku, drukarka do drukowania wzoru czarno-białego lub skali szarości na płycie papierowej, brzęczyk lub głośnik

Maszyna sortująca	Odtwarzacz płyt świetlnych
Hub Bariera świetlna Wskaźnik stanu LED Jasność/kolor Wyrzutnik do sortowania	Stolik z papierowymi nakładkami z czarno-białymi wzorami lub szarościami Czujnik światła jako głowica pomiarowa Głośnik (nie wchodzi w skład zestawu materiałowego)

Format: Samodzielna praca uczniów

Treść:

- Sortownica: Użyj wyrzutnika do sortowania różnego rodzaju przedmiotów, które różnią się np. jasnością lub kolorem.



- Odtwarzacz płyt świetlnych: jeśli na stoliku umieścimy papier z czarno-białym wzorem, czujnik światła może być wykorzystany do przetworzenia informacji o jasności na tony i proste rytmy dla prostego urządzenia wyjściowego dźwięku. Do tego celu można również wykorzystać np. wydrukowane skale szarości.



Działania uczniów

- Młodzi ludzie programują mini linię produkcyjną jako maszynę do sortowania i rozróżniają kubki według koloru lub jasności.
- Uczniowie projektują wzory w skali szarości dla stolika. Arduino rozpoznaje wartości jasności i odtwarza odpowiednie dźwięki. Młodzież może również sama przypisać dźwięki do wartości jasności w programowaniu.

4.12 Ogólne uwagi dotyczące wideokonferencji i pracy praktycznej w sytuacjach na odległość

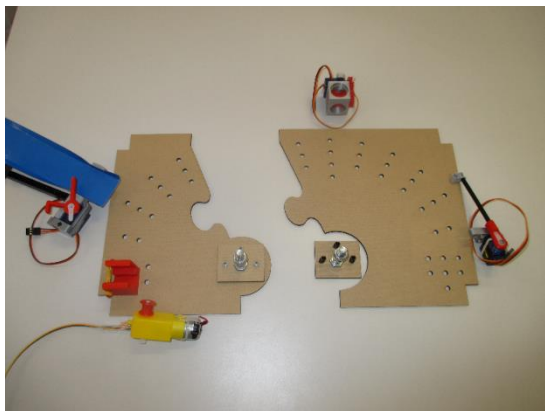
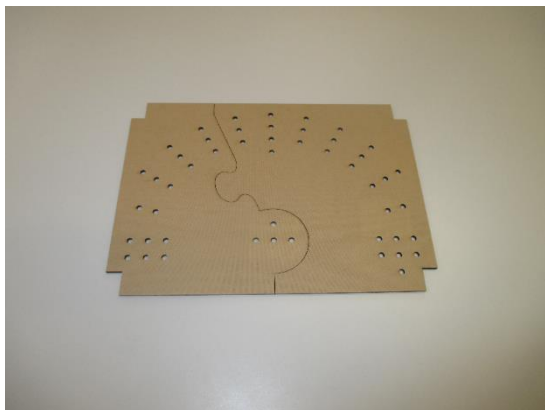
Ochrona danych

Szkoła powinna wyjaśnić kwestie dotyczące niezbędnego przetwarzania danych podczas wideokonferencji – np. stosowania współdzielenia ekranu lub korzystania z kamery – zgodnie z ogólnym rozporządzeniem o ochronie danych. W ścisłym porozumieniu z uczniami, nauczyciele mogą prowadzić lekcje bez włączonej kamery. Jednak gdy uczniowie włączają kamery, znacznie łatwiej jest dostrzec postępy i może to znacznie poprawić samopoczucie w grupie. Jedną z opcji, którą mogą zaproponować nauczyciele, jest ustawienie kamery wyłącznie na eksperymenty, bez konieczności umieszczania na zdjęciu samych młodych ludzi – ale ta opcja nie jest dostępna dla wszystkich ze względu na kwestie związane z technologią kamery i umieszczeniem eksperymentów.

Gospodarka materiałowa

Pierwsza część sekwencji projektu, tj. sekcje od 1.a Wprowadzenie do tematu do 2.b Serwomotor, może być łatwo przeprowadzona w środowisku nauczania online. W tym celu nauczyciele potrzebują odpowiedniego systemu wideokonferencyjnego, który umożliwi im pokazanie prezentacji online, a także, w razie potrzeby, zorganizowanie salek, w których uczniowie mogą pracować niezależnie w zespołach. Część materiałów praktycznych, których młodzi ludzie potrzebują do pracy w domu w tej części, składa się z Arduino, płytki z wtyczkami, kabli ze zworkami, modułu świateł drogowych lub diod LED i serwomechanizmów – innymi słowy, ogólnie rzecz biorąc, stosunkowo łatwy do opanowania sprzęt. W sytuacji pełnego dystansu wymaga to jednak, aby wszyscy młodzi ludzie w klasie otrzymali taki zestaw. Jeśli nie jest to możliwe, nauczyciel może zamiast tego pogłębić pracę za pomocą internetowego środowiska symulacyjnego Wokwi. Może się to jednak odbyć kosztem nabycia umiejętności niezbędnych w późniejszych jednostkach w zakresie obsługi mikrokontrolera.

Jeśli chodzi o stół obrotowy, realizacja jest nieco bardziej skomplikowana. Najlepiej byłoby, gdyby wszyscy uczniowie otrzymali komplet materiałów do pracy w domu. Jeśli nie jest to możliwe (ze względu na koszty, wysiłek związany z produkcją materiałów lub trudności z ich wypożyczeniem lub przesłaniem), potrzebna jest alternatywa.



Alternatywą może być na przykład podzielenie płyty bazowej dla stołu i danie uczniom różnych części, tak aby mogli pracować razem jako zespół przy projektowaniu procesu automatyzacji.

5

Możliwości zastosowania modułowego

W zależności od danych warunków pomocna może być możliwość wykorzystania modułu dydaktycznego w innej formie. W dalszej części przedstawiono dwie alternatywne procedury; istnieje również wiele innych możliwości.

Procedura 1

W centrum sekwencji 1 nadal znajduje się proces automatyzacji. Jednak proces ten został zredukowany do transportu kubków za pomocą stołu obrotowego, zastosowania serwomotoru do dozownika oraz zależnego od czujników sterowania procesem produkcji.

1.1 Wprowadzenie do zastosowania stolików obrotowych w automatyce oraz cel miniaturyzacji procesu automatyzacji.
1.2 Wprowadzenie do pracy z Arduino z wykorzystaniem pierwszego projektu obwodu i przykładu programowania z różnokolorowymi diodami LED.
2.1 Wprowadzenie do serwomotorów, w tym obwody i programowanie prawdziwego serwomotoru. Wykorzystanie serwomotoru do pracy z dozownikiem.
2.2 Budowa stolika, układy i programowanie silnika napędowego.
2.3 Wykorzystanie czujnika światła wraz z układem, programowaniem i wyświetlaniem zmierzonych wartości za pomocą plotera. Wykorzystanie czujnika światła do zależnego od czujnika sterowania "procesem produkcyjnym".
3. automatyka w kontekście społecznym

Procedura 2

Przy sekwencji 2 stół jest wykorzystywany w zupełnie inny sposób, a mianowicie wyłącznie jako odtwarzacz płyt świetlnych. W związku z tym konieczne są tylko podstawowe czynności związane z pracą z Arduino, wprawianiem stołu w ruch i wstawianiem fotokomórek.

1.1 Wprowadzenie do samodzielnie wybranego tematu
1.2 Wprowadzenie do pracy z Arduino z wykorzystaniem pierwszego projektu obwodu i przykładu programowania z różnymi kolorowymi diodami LED
2.1 Budowa stolika, układy i programowanie silnika napędowego.
2.2 Obsługa fotokomórki, w tym obwody, programowanie i wyświetlanie wartości pomiarowych za pomocą plotera.
3) wykorzystanie stołu jako odtwarzacza płyt świetlnych

6 Dodatek: Wgląd w interaktywną instrukcję multimedialną strona

Zrzuty ekranu przedstawiają przykładową instrukcję krok po kroku ze zdjęciami animacji, filmów i efektów mouseover do ustawiania, okablowania i programowania zasuwy wejściowej i bariery refleksyjnej. Aby uzyskać lepsze wrażenie, warto powiększyć widok.

Automatyka w miniaturze

Co-funded by the Erasmus+ Programme of the European Union

Partnerzy projektu europejskiego:

Deutsches Museum, LISBOA, Technische Universität München, CENTRUM NAUKI KOPERNIK, Copernicus Science Center, BS Info, BS Fisi

we współpracy z:

Przegląd

Automatyzacja z Arduino - Stacja napelniania linii produkcyjnej

Przesuń myszką na pola

```

    graph LR
      Rozdziel[rozdzielić kubki] --> Wykryj[wykryć kubki]
      Wykryj --> Koncowy[koncowy]
      Koncowy --> Transport2[transport]
      Transport2 --> Wypelni[wypełnić]
      Wypelni --> Transport1[transport kubka]
      Transport1 --> Rozdziel
  
```

Zadania: Kliknij na pole i przeskocz do odpowiedniego rozdziału.
Skończyłeś już wszystko? W takim razie kliknij tutaj: [zakończzone](#)

Przegląd

Automatyzacja z Arduino - Stacja napelniania linii produkcyjnej

Przesuń myszką na pola

```

    graph LR
      Rozdziel[rozdzielić kubki] --> Wykryj[wykryć kubki]
      Wykryj --> Koncowy[koncowy]
      Koncowy --> Transport2[transport]
      Transport2 --> Wypelni[wypełnić]
      Wypelni --> Transport1[transport kubka]
      Transport1 --> Rozdziel
  
```

Zadania: Kliknij na pole i przeskocz do odpowiedniego rozdziału.
Skończyłeś już wszystko? W takim razie kliknij tutaj: [zakończzone](#)

Sprzęt – prowadnica ślizgowa

Prowadnica ślizgowa sprawia, że kubki przesuwają się jeden po drugim na stole obrotowym. W tym celu serwonmotor obraca barierę tam i z powrotem pomiędzy dwoma pozycjami. Dzięki temu kubek nie musi być umieszczany na płycie ręcznie.

Przesuń myszką nad obrazkami...

W dalszej części należy najpierw zamontować prowadnicę ślizgową, następnie podłączyć serwonmechanizm do Arduino i na koniec zaprogramować Arduino.

Hands-on-Remote

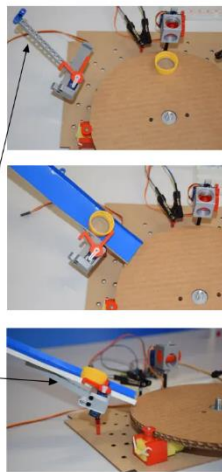
wcześniej
struktura
stolik obrotowy
dozownik
bariera świetlna
wyrzutnik wyjściowy
przewodnica ślizgowa

przegląd

Sprzęt - montaż przewodnicy ślizgowej

Zamontować przewodnicę ślizgową z bramką:

1. Zamocować przewodnicę ślizgową z podstawą w płycie podstawowej.
2. Wsunąć przewodnicę
3. Ramię (belka) Lego służy jako podpora dla zjeżdżalni.

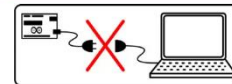


Hands-on-Remote

wcześniej
struktura
stolik obrotowy
dozownik
bariera świetlna
wyrzutnik wyjściowy
przewodnica ślizgowa

przegląd

Okablowanie - przewodnica wejściowa



Przed zmianą okablowania proszę odłączyć Arduino od komputera!

Pracuj na Arduino tylko w stanie wyłączonym!

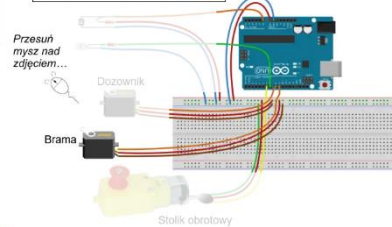
Teraz do Arduino należy podłączyć serwo bramki na przewodnicę ślizgową:

1. Użyj magistrali zasilającej na płycie prototypowej (czerwonej i niebieskiej):

czerwony przewód: na Arduino V in - Pin -> na płycie prototypowej czerwona szyna zasilania
niebieski przewód: GND - pin -> niebieska magistrala masy

2. Podłączyć serwo przewodnicy ślizgowej (bramy) do:

Pomarańczowy -> Pin 9
czerwony -> V in - magistrala
Brazowy -> GND - magistrala

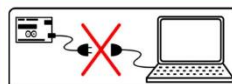


Hands-on-Remote

wcześniej
struktura
stolik obrotowy
dozownik
bariera świetlna
wyrzutnik wyjściowy
przewodnica ślizgowa

przegląd

Okablowanie - przewodnica wejściowa



Przed zmianą okablowania proszę odłączyć Arduino od komputera!

Pracuj na Arduino tylko w stanie wyłączonym!

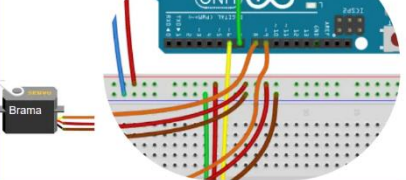
Teraz do Arduino należy podłączyć serwo bramki na przewodnicę ślizgową:

1. Użyj magistrali zasilającej na płycie prototypowej (czerwonej i niebieskiej):

czerwony przewód: na Arduino V in - Pin -> na płycie prototypowej czerwona szyna zasilania
niebieski przewód: GND - pin -> niebieska magistrala masy

2. Podłączyć serwo przewodnicy ślizgowej (bramy) do:

Pomarańczowy -> Pin 9
czerwony -> V in - magistrala
Brazowy -> GND - magistrala



Hands-on-Remote

wcześniej
struktura
stolik obrotowy
dozownik
bariera świetlna
wyrzutnik wyjściowy
przewodnica ślizgowa

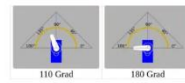
przegląd

Sprzęt - Bramka-Servo

Barierka przewodnicy ślizgowej (=bramy) jest sterowana przez serwo mechanizm.

Serwo mechanizmy to małe silniczki, które zazwyczaj mogą .

Przesuń myszką nad obrazkami...



Dla zlabanu barierki potrzebujemy dwóch pozycji:
-> bariera zamknięta (= pozycja spoczynkowa, zatrzymuje wszystkie kubki.
-> bariera otwarta (= pozycja przejścia, jeden kubek wsuwa się na tarczę)

Skośny wskaźnik...



Kliknij na obrazek aby rozpocząć animację...



Dokładne ustawienie kąta obu pozycji należy ustalić metodą prób i błędów.

Na marginesie: Wartość kąta pozycji spoczynkowej jest większa od wartości kąta pozycji przełotowej.

Hands-on-Remote

wcześniej
struktura
stolik obrotowy
dozownik
bariera świetlna
wyrzutnik wyjściowy
przewodnica ślizgowa

przegląd

Sprzęt - Bramka-Servo

Barierka przewodnicy ślizgowej (=bramy) jest sterowana przez serwo mechanizm.

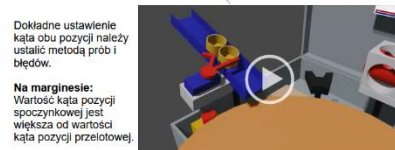
Serwo mechanizmy to małe silniczki, które zazwyczaj mogą .

Serwo mechanizmy to małe silniczki, które zazwyczaj mogą .

Dla zlabanu barierki potrzebujemy dwóch pozycji:
-> bariera zamknięta (= pozycja spoczynkowa, zatrzymuje wszystkie kubki.
-> bariera otwarta (= pozycja przejścia, jeden kubek wsuwa się na tarczę)



Kliknij na obrazek aby rozpocząć animację...



Dokładne ustawienie kąta obu pozycji należy ustalić metodą prób i błędów.

Na marginesie: Wartość kąta pozycji spoczynkowej jest większa od wartości kąta pozycji przełotowej.

Hands-on-Remote

wcześniej
struktura
stolik obrotowy
dozownik
bariera świetlna
wyrzutnik wyjściowy
przewodnica ślizgowa

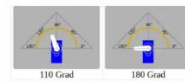
przegląd

Sprzęt - Bramka-Servo

Barierka przewodnicy ślizgowej (=bramy) jest sterowana przez serwo mechanizm.

Serwo mechanizmy to małe silniczki, które zazwyczaj mogą .

Przesuń myszką nad obrazkami...



Dla zlabanu barierki potrzebujemy dwóch pozycji:
-> bariera zamknięta (= pozycja spoczynkowa, zatrzymuje wszystkie kubki.
-> bariera otwarta (= pozycja przejścia, jeden kubek wsuwa się na tarczę)

Skośny wskaźnik bramki zapobiega przeslizgiwaniu się kolejnych kubków podczas otwierania zlabanu.



Kliknij na obrazek aby rozpocząć animację...



Dokładne ustawienie kąta obu pozycji należy ustalić metodą prób i błędów.

Na marginesie: Wartość kąta pozycji spoczynkowej jest większa od wartości kąta pozycji przełotowej.

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Bramka sterująca



Praca w zakładce ino.

Zadanie: Zaprogramuj serwomechanizm bramy tak, aby mógł wyprowadzić kubek. Znajdź odpowiednie wartości kątów dla pozycji spoczynkowej i przełotowej.



1. Dołącz plik "h", którego używasz, za pomocą instrukcji #include. ...
2. Utwórz obiekt serwa, nadaj mu nazwę i ...
3. W setup() ...
4. Obróć bramkę w loop() ...
5. Skomentuj swój program...

```
sketch-OOP $ include
/*
 *
 */
#include "tunlab.h"

ServoMotor name(pin_pos0_pos1);

void setup() {
  //
  .go(LOW);
}

void loop() {
  //
  .goSlow(HIGH);
}
}
```



Zadanie dodatkowe: dioda LED powinna wskazywać aktywność bramki (całkowity proces: obrót do pozycji przełotowej i powrót do pozycji spoczynkowej).

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Bramka sterująca



Praca w zakładce ino.

Zadanie: Zaprogramuj serwomechanizm bramy tak, aby mógł wyprowadzić kubek. Znajdź odpowiednie wartości kątów dla pozycji spoczynkowej i przełotowej.



1. Dołącz plik "h", którego używasz, za pomocą instrukcji #include. ...
2. Utwórz obiekt serwa, nadaj mu nazwę i ...
3. W setup() ...
4. Obróć bramkę w loop() ...
5. Skomentuj swój program...

```
sketch-OOP $ include
/*
 *
 */
#include "tunlab.h"

ServoMotor name(pin_pos0_pos1);

void setup() {
  //
  .go(LOW);
}

void loop() {
  //
  .goSlow(HIGH);
}
}
```



Zadanie dodatkowe: dioda LED powinna wskazywać aktywność bramki (całkowity proces: obrót do pozycji przełotowej i powrót do pozycji spoczynkowej).

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Bramka sterująca



Praca w zakładce ino.

Zadanie: Zaprogramuj serwomechanizm bramy tak, aby mógł wyprowadzić kubek. Znajdź odpowiednie wartości kątów dla pozycji spoczynkowej i przełotowej.



1. Dołącz plik "h", którego używasz, za pomocą instrukcji #include. ...
2. Utwórz obiekt serwa, nadaj mu nazwę i podaj numer pinu, w którym podłączyłeś serwo oraz dwie pozycje kątowe w nawiasach ().
3. W setup() ...
4. Obróć bramkę w loop() ...
5. Skomentuj swój program...

```
sketch-OOP $ include
/*
 *
 */
#include "tunlab.h"

ServoMotor gate(9,100,10);

void setup() {
  //
  .go(LOW);
}

void loop() {
  //
  .goSlow(HIGH);
}
}
```



Zadanie dodatkowe: dioda LED powinna wskazywać aktywność bramki (całkowity proces: obrót do pozycji przełotowej i powrót do pozycji spoczynkowej).

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Bramka sterująca



Praca w zakładce ino.

Zadanie: Zaprogramuj serwomechanizm bramy tak, aby mógł wyprowadzić kubek. Znajdź odpowiednie wartości kątów dla pozycji spoczynkowej i przełotowej.



1. Dołącz plik "h", którego używasz, za pomocą instrukcji #include. ...
2. Utwórz obiekt serwa, nadaj mu nazwę i ...
3. W setup() można doprowadzić serwo do pozycji spoczynkowej metodą go().
4. Obróć bramkę w loop() ...
5. Skomentuj swój program...

```
sketch-OOP $ include
/*
 *
 */
#include "tunlab.h"

ServoMotor gate(9,100,10);

void setup() {
  //
  gate.go(LOW);
}

void loop() {
  //
  .goSlow(HIGH);
}
}
```



Zadanie dodatkowe: dioda LED powinna wskazywać aktywność bramki (całkowity proces: obrót do pozycji przełotowej i powrót do pozycji spoczynkowej).

Hands-on-Remote

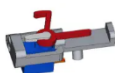
- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Bramka sterująca



Praca w zakładce ino.

Zadanie: Zaprogramuj serwomechanizm bramy tak, aby mógł wyprowadzić kubek. Znajdź odpowiednie wartości kątów dla pozycji spoczynkowej i przełotowej.



1. Dołącz plik "h", którego używasz, za pomocą instrukcji #include. ...
2. Utwórz obiekt serwa, nadaj mu nazwę i ...
3. W setup() ...
4. Obróć bramkę w loop() za pomocą metody go() w pozycję pass i z powrotem przekazując odpowiednio wartości 'HIGH' i 'LOW'.
5. Skomentuj swój program...

```
sketch-OOP $ include
/*
 *
 */
#include "tunlab.h"

ServoMotor gate(9,100,10);

void setup() {
  //
  gate.go(LOW);
}

void loop() {
  //
  gate.goSlow(HIGH);
  delay(1000);
  gate.goSlow(LOW);
  delay(2000);
}
}
```



Zadanie dodatkowe: dioda LED powinna wskazywać aktywność bramki (całkowity proces: obrót do pozycji przełotowej i powrót do pozycji spoczynkowej).

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Bramka sterująca



Praca w zakładce ino.

Zadanie: Zaprogramuj serwomechanizm bramy tak, aby mógł wyprowadzić kubek. Znajdź odpowiednie wartości kątów dla pozycji spoczynkowej i przełotowej.



1. Dołącz plik "h", którego używasz, za pomocą instrukcji #include. ...
2. Utwórz obiekt serwa, nadaj mu nazwę i ...
3. W setup() ...
4. Obróć bramkę w loop() ...
5. Skomentuj swój program: komentarze wielowierszowe zaczynają się od /*, a kończą na */ // wprowadza komentarz jednowierszowy.

```
sketch-OOP $ include
/*
 * a cup slips through
 * 25.01.2022 Mike
 */
#include "tunlab.h"

ServoMotor gate(9,100,10);

void setup() {
  //
  // ejector in rest position:
  gate.go(LOW);
}

void loop() {
  //
  // ejector in action:
  gate.goSlow(HIGH);
  delay(1000);
  gate.goSlow(LOW);
  delay(2000);
}
}
```



Zadanie dodatkowe: dioda LED powinna wskazywać aktywność bramki (całkowity proces: obrót do pozycji przełotowej i powrót do pozycji spoczynkowej).

Hands-on-Remote

wcześniej

struktura

stolik obrotowy

dozownik

bariera świetlna

wyrzutnik wyjściowy

przewodnica ślizgowa

przegląd

Przegląd

Automatyzacja z Arduino -
Stacja napełniania linii produkcyjnej

Przesuń myszką na pola

Rozdzielić kubki

transport kubka

wykręć kubki

wypełnić

końcowy

transport

Zadania: Kliknij na pole i przeskocz do odpowiedniego rozdziału.

Skończyłeś już wszystko? W takim razie kliknij tutaj: [zakończ](#)

Hands-on-Remote

wcześniej

struktura

stolik obrotowy

dozownik

bariera świetlna

wyrzutnik wyjściowy

przewodnica ślizgowa

przegląd

Sprzęt - LED i LDR

Czujnik optyczny, np. nasz LDR, oraz źródło światła, np. nasza dioda LED, współpracują ze sobą tworząc barierę świetlną. W przeciwieństwie do wielu innych barier świetlnych, wiązka światła nie jest przerywana. Nasza bariera świetlna działa na zasadzie odbicia wiązki światła, jest to bariera refleksyjna.

LED i LDR celują w ten sam punkt.

Bariera świetlna składa się z dwóch aktywnych elementów ...

Jesli kubek przesunie się w stożek świetlny diody ...

Dioda LED oświetla ...

LDR mierzy ...

Zwróć uwagę na LDR ↓

Hands-on-Remote

wcześniej

struktura

stolik obrotowy

dozownik

bariera świetlna

wyrzutnik wyjściowy

przewodnica ślizgowa

przegląd

Sprzęt - LED i LDR

Czujnik optyczny, np. nasz LDR, oraz źródło światła, np. nasza dioda LED, współpracują ze sobą tworząc barierę świetlną. W przeciwieństwie do wielu innych barier świetlnych, wiązka światła nie jest przerywana. Nasza bariera świetlna działa na zasadzie odbicia wiązki światła, jest to bariera refleksyjna.

LED i LDR celują w ten sam punkt.

Bariera świetlna składa się z dwóch aktywnych elementów ...

Jesli kubek przesunie się w stożek świetlny diody, odbija jej światło, a LDR rejestruje wzrost jasności.

Dioda LED oświetla ...

LDR mierzy ...

Zwróć uwagę na LDR ↓

Hands-on-Remote

wcześniej

struktura

stolik obrotowy

dozownik

bariera świetlna

wyrzutnik wyjściowy

przewodnica ślizgowa

przegląd

Sprzęt - Bariera świetlna

Kolejny krok w konstrukcji mechanicznej: Refleksyjna bariera świetlna. Kubek na się niezawodnie zatrzymać przed dozownikiem. W tym celu czujnik optyczny wykrywa jego położenie.

Przesuń myszką nad obrazkami...

W dalszej części artykułu najpierw zamontuj fotokomórkę, następnie podłącz do Arduino zarówno LDR (czujnik światła) jak i diodę LED (źródło światła), a na koniec zaprogramuj Arduino.

Hands-on-Remote

wcześniej

struktura

stolik obrotowy

dozownik

bariera świetlna

wyrzutnik wyjściowy

przewodnica ślizgowa

przegląd

Sprzęt - Montaż barier światłowej

Zamontuj barierę światłową reflex: Uchwyt barier światłowej można zamocować w płycie podstawowej za pomocą podstawki.

Użyj ogranicznika osi, aby ustawić barierę światłową na odpowiedniej wysokości.

LDR i LED są już przyłutowane Z...

Uwaga: Podłącz pojedynczą ciągłą nóżkę do...

Hands-on-Remote

40

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Okablowanie - bariera świetlna

Przed zmianą okablowania proszę odłączyć Arduino od komputera!

Pracuj na Arduino tylko w stanie wyłączonym!

Teraz połącz LDR (czujnik światła) i LED (źródło światła) do Arduino.

Ponownie użyj szyny zasilającej (czerwonej i niebieskiej) na płycie prototypowej:

- na Arduino: V in - Pin, GND - pin
- na płycie prototypowej: czerwona szyna zasilania, niebieska magistrala masy

1. Podłącz LDR:

- Złoty -> Pin A0
- Zielony -> szyna zasilania
- Niebieski -> magistrala GND

Przesuń mysz nad zdjęciem

2. Podłącz diodę LED:

- Szary -> Pin 7
- Czarny -> GND (nie jest potrzebny biały przewód do diody LED).

Stolik obrotowy

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Okablowanie - bariera świetlna

Przed zmianą okablowania proszę odłączyć Arduino od komputera!

Pracuj na Arduino tylko w stanie wyłączonym!

Teraz połącz LDR (czujnik światła) i LED (źródło światła) do Arduino.

Ponownie użyj szyny zasilającej (czerwonej i niebieskiej) na płycie prototypowej:

- na Arduino: V in - Pin, GND - pin
- na płycie prototypowej: czerwona szyna zasilania, niebieska magistrala masy

1. Podłącz LDR:

- Złoty -> Pin A0
- Zielony -> szyna zasilania
- Niebieski -> magistrala GND

2. Podłącz diodę LED:

- Szary -> Pin 7
- Czarny -> GND (nie jest potrzebny biały przewód do diody LED).

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Wartości pomiarowe - światło odbite i światło otoczenia

Jeśli wiązka światła diody LED zmierza donikąd (brak kubka przed fotokomórką), jej światło nie jest odbijane, czujnik światła widzi tylko światło otoczenia.

Jeśli w wiązce światła znajduje się kubek, światło diody LED odbite przez kubek dodaje się do światła otoczenia, czujnik światła mierzy większą jasność.

Uwaga: im większa jasność, tym mniejsza wartość pomiaru. Wynika to ze sposobu podłączenia rezystora szeregowego i czujnika światła.

Niebieska linia: Wartości pomiarowe przy włączonej diodzie LED
Linia czerwona: Wartości pomiarowe przy wyłączonej diodzie LED
Zielona linia: Różnica pomiędzy dwoma zmierzonymi wartościami

Dopiero gdy kubek przejdzie przed barierą świetlną i odbije światło, oba wartości różnią się znacząco. Zielona linia pokazuje różnicę zmierzonych wartości. Oceniamy tę różnicę.

Jeśli różnica przekracza ustaloną przez nas wartość, kubek uznaje się za wykryty.

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Klasa "ReflexSensor"

Podczas tworzenia obiektu klasa "ReflexSensor" oczekuje ...

Możliwe jest również wykorzystanie samego LDR. W tym przypadku ...

Metody showLDR() i showValues() mogą być użyte do ...

Klasa "ReflexSensor" udostępnia metodę "waitForCan()", która ...

```

class ReflexSensor {
public:
    • name: String
    • pinLEDnumber: int
    • pinLDRnumber: int
    • trigger: int

    • ReflexSensor(pinLED, pinLDR, trigger)
    • ReflexSensor(pinLDR)
    • showLDR(void): void
    • showValues(void): void
    • waitForCan(void): void
    • detectCan(void): bool
};
    
```

```

// create light sensor object
ReflexSensor xxx (A0);
ReflexSensor yyy (7, A0, 30);

void setup() {
  xxx.showLDR(); //
  yyy.showValues();
}

void loop() {
  xxx.waitForCan();
  if (xxx.detectCan() == HIGH) {
    //
  }
}
    
```

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Klasa "ReflexSensor"

Podczas tworzenia obiektu klasa "ReflexSensor" oczekuje, oprócz nazwy, numerów pinów, do których podłączone są diody LED i LDR oraz wartości progowej, od której uznaje się, że kubek został wykryty.

Możliwe jest również wykorzystanie samego LDR. W tym przypadku ...

Metody showLDR() i showValues() mogą być użyte do ...

Klasa "ReflexSensor" udostępnia metodę "waitForCan()", która ...

```

// create light sensor object
ReflexSensor xxx (A0);
ReflexSensor yyy (7, A0, 30);

void setup() {
  xxx.showLDR(); //
  yyy.showValues();
}

void loop() {
  xxx.waitForCan();
  if (xxx.detectCan() == HIGH) {
    //
  }
}
    
```

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Klasa "ReflexSensor"

Podczas tworzenia obiektu klasa "ReflexSensor" oczekuje ...

Możliwe jest również wykorzystanie samego LDR. W tym przypadku oprócz nazwy wymagany jest tylko numer pinu LDR.

Metody showLDR() i showValues() mogą być użyte do ...

Klasa "ReflexSensor" udostępnia metodę "waitForCan()", która ...

```

// create light sensor object
ReflexSensor xxx (A0);
ReflexSensor yyy (7, A0, 30);

void setup() {
  xxx.showLDR(); //
  yyy.showValues();
}

void loop() {
  xxx.waitForCan();
  if (xxx.detectCan() == HIGH) {
    //
  }
}
    
```


Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Klasa "ReflexSensor"

ReflexSensor

- name: String
- pinLEDNumber: int
- pinLDRnumber: int
- trigger: int
- ReflexSensor(pinLED, pinLDR, trigger)
- ReflexSensor(pinLDR)
- showLDR(void): void
- showValues(void): void
- waitForCan(void): void
- detectCan(void): bool

Przesuń mysz nad wiersze tekstu...

Podczas tworzenia obiektu klasa "ReflexSensor" oczekuje ...

Możliwe jest również wykorzystanie samego LDR. W tym przypadku ...

Metody showLDR() i showValues() mogą być użyte do graficznego wyświetlenia wartości pomiarowych zwróconych przez LDR.

Klasa "ReflexSensor" udostępnia metodę "waitForCan()", która ...

```

// create light sensor objects:
ReflexSensor xxx (A0);
ReflexSensor yyy (7, A0, 30);

void setup() {
  xxx.showLDR(); //
  yyy.showValues();
}

void loop() {
  xxx.waitForCan();
  if (xxx.detectCan() == HIGH) {
    //
  }
}

```

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Klasa "ReflexSensor"

ReflexSensor

- name: String
- pinLEDNumber: int
- pinLDRnumber: int
- trigger: int
- ReflexSensor(pinLED, pinLDR, trigger)
- ReflexSensor(pinLDR)
- showLDR(void): void
- showValues(void): void
- waitForCan(void): void
- detectCan(void): bool

Przesuń mysz nad wiersze tekstu...

Podczas tworzenia obiektu klasa "ReflexSensor" oczekuje ...

Możliwe jest również wykorzystanie samego LDR. W tym przypadku ...

Metody showLDR() i showValues() mogą być użyte do ...

Klasa "ReflexSensor" udostępnia metodę "waitForCan()", która zatrzymuje przepływ programu do momentu wykrycia kubka.

```

// create light sensor objects:
ReflexSensor xxx (A0);
ReflexSensor yyy (7, A0, 30);

void setup() {
  xxx.showLDR(); //
  yyy.showValues();
}

void loop() {
  xxx.waitForCan();
  if (xxx.detectCan() == HIGH) {
    //
  }
}

```

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Klasa "ReflexSensor"

ReflexSensor

- name: String
- pinLEDNumber: int
- pinLDRnumber: int
- trigger: int
- ReflexSensor(pinLED, pinLDR, trigger)
- ReflexSensor(pinLDR)
- showLDR(void): void
- showValues(void): void
- waitForCan(void): void
- detectCan(void): bool

Przesuń mysz nad wiersze tekstu...

Podczas tworzenia obiektu klasa "ReflexSensor" oczekuje ...

Możliwe jest również wykorzystanie samego LDR. W tym przypadku ...

Metody showLDR() i showValues() mogą być użyte do ...

Klasa "ReflexSensor" udostępnia metodę "waitForCan()", która ...

```

// create light sensor objects:
ReflexSensor xxx (A0);
ReflexSensor yyy (7, A0, 30);

void setup() {
  xxx.showLDR(); //
  yyy.showValues();
}

void loop() {
  xxx.waitForCan();
  if (xxx.detectCan() == HIGH) {
    //
  }
}

```

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Ustawienie bariery świetlnej

Czułość bariery świetlnej jest określona przez wartość wyzwalacza.

Aby poznać wartość wyzwalającą, można pokazać zmierzone wartości czujnika światła na wykresie plotera.

Praca w zakładce Ino.

- 1) Utwórz nowy obiekt klasy ReflexSensor o nazwie, pinie i wartości wyzwalacza. Na razie użyj wartości 30.
- 2) Określ wartość wyzwalacza:...
- 3) Otwórz ploter (kreslarka)...
- 4) I wyznacz ...
- 5) Uwaga: ...
- 6) Teraz dodaj ...

Proszę wprowadzić założoną wartość 30 jako wartość wyzwalającą i dopasować wartość w kroku (4).

```

// Extension of the programme code
ReflexSensor lightBarrier (7, A0, 30);

void setup() {
  //
}

void loop() {
  //
}

```

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Ustawienie bariery świetlnej

Czułość bariery świetlnej jest określona przez wartość wyzwalacza.

Aby poznać wartość wyzwalającą, można pokazać zmierzone wartości czujnika światła na wykresie plotera.

Praca w zakładce Ino.

- 1) Utwórz nowy obiekt...
- 2) Określ wartość wyzwalacza:...
- 3) Otwórz ploter (kreslarka) Niech kubki przejdą przez barierę świetlną.
- 4) I wyznacz ...
- 5) Uwaga: ...
- 6) Teraz dodaj ...

```

// Extension of the programme code
ReflexSensor lightBarrier (7, A0, 30);

void setup() {
  //
}

void loop() {
  //
}

```

Hands-on-Remote

- wcześniej
- struktura
- stolik obrotowy
- dozownik
- bariera świetlna
- wyrzutnik wyjściowy
- przewodnica ślizgowa
- przegląd

Programowanie: Ustawienie bariery świetlnej

Czułość bariery świetlnej jest określona przez wartość wyzwalacza.

Aby poznać wartość wyzwalającą, można pokazać zmierzone wartości czujnika światła na wykresie plotera.

Praca w zakładce Ino.

- 1) Utwórz nowy obiekt...
- 2) Określ wartość wyzwalacza:...
- 3) Otwórz ploter (kreslarka)...
- 4) I wyznacz pasującą wartość wyzwalacza z plotera (kreslarka) i załóż nim 30.
- 5) Uwaga: ...
- 6) Teraz dodaj ...

Wartość wyzwalająca

Kubki przechodzą przez barierę świetlną

```

// Extension of the programme code
ReflexSensor lightBarrier (7, A0, 30);

void setup() {
  //
}

void loop() {
  //
}

```



**CENTRUM
NAUKI
KOPERNIK**



Technische Universität München

Wszystkie materiały dostępne są na stronie

<https://erasmus-plus.ec.europa.eu/projects/search/details/2020-1-DE02-KA226-VET-008295>

Projekt Hands-on Remote był finansowany w ramach programu Erasmus+ KA226 Partnerships for Digital Education Readiness (2020-1-DE02-KA226-VET-008295).

Co-funded by the
Erasmus+ Programme
of the European Union

