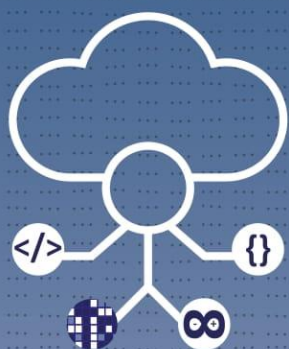


Mikrokontrolery **MEGA**projekty

Podręcznik dla prowadzących zajęcia interdyscyplinarne
z pomocą mikrokontrolerów



Mikrokontrolery **MEGA**projekty

Damian Kimła
Dariusz Świder

Prawa autorskie:

1. Mikrokontrolery – megaprojekty by Damian Kimla i Dariusz Świder w ramach POPOJUTRZE 2.0 – KSZTAŁCENIE <https://popojutrze2.pl> CC BY-SA 4.0 zrealizowanych przez Zakład Doskonalenia Zawodowego w Katowicach – w ramach projektu realizowanego przez SENSE Consulting sp. z o. o.
2. Majątkowe prawa autorskie do udostępnionego utworu należą do Ministra Funduszy i Polityki Regionalnej (dalej zwany „Licencjodawcą”), który udzielił udostępniającemu niniejszy utwór licencji Creative Commons: Uznanie autorstwa- Na tych samych warunkach 4.0, tj. nieodpłatnej, nieobejmującej prawa do udzielania sublicencji, niewyłącznej, nieodwoławnej licencji na korzystanie z Utworu na terytorium całego świata, tj. do zwielokrotniania i dzielenia się utworem w całości i części, a także tworzenia i zwielokrotniania i dzielenia się utworami zależnymi do tego utworu (dalej zwana „Licencją”).
3. Treść Licencji jest dostępna na stronie <http://creativecommons.org/licenses/by-sa/4.0/legalcode.pl>.
4. Jeżeli Licencjodawca oddzielnie nie postanowił inaczej, Licencjodawca, w możliwie najszerszym zakresie, oferuje utwór licencjonowany w takiej formie, w jakiej zapoznał się z nim licencjobiorca i nie udziela żadnych zapewnień, ani jakiegokolwiek rodzaju gwarancji, dotyczących utworu licencjonowanego, ani wynikających z wyraźnego postanowienia, dorozumianych, ustawowych, ani jakichkolwiek innych. Obejmuje to, bez ograniczeń, rękojmię, zbywalność, przydatność do konkretnego celu, brak naruszeń praw innych osób, brak ukrytych lub innych wad, dokładność, występowanie lub niewystępowanie wad widocznych jak i ukrytych. W przypadku, gdy wyłączenie gwarancji nie jest dozwolone w całości lub w części, niniejsze wyłączenie może nie mieć zastosowania do licencjobiorcy.

Spis treści

1	Wstęp	4
2	Wskazówki dotyczące modyfikacji zajęć oraz kodów źródłowych.....	6
2.1	Temat: Kolonializm i jego upadek	6
2.1.1.	Wskazówki modyfikacji kodu źródłowego	10
2.1.2.	Modyfikacje programu, potrzebne do dodania dodatkowego (piątego) połączenia pinów mikrokontrolera.....	12
2.2	Temat: Rozpady promieniotwórcze.....	14
2.3	Temat: Masa powietrza w sali lekcyjnej.....	16
3	Opis dodatkowych 5 elementów uatrakcyjnających zajęcia na przykładzie tematu Kolonializm i jego upadek	21
3.1.	Wyświetlacz LCD.....	21
3.2.	Diody LED.....	23
3.3.	Przycisk	24
3.4.	Fotorezystor	26
3.5.	Buzzer.....	28
3.6.	Łączenie poszczególnych wariantów	29

1 Wstęp

*„...należy zmienić myślenie o przekazywaniu wiedzy, roli nauczyciela i ucznia.
Należy zacząć prowadzić zajęcia w duchu projektowym, a nie transmisyjnym...”*

Cytat zaczerpnięty z nauczycielskiego forum internetowego

*„W XXI wieku umiejętnością, która jest niezwykle pożądana u uczniów,
jest przede wszystkim kreatywność.*

*Innowacyjne podejście do pojawiających się wyzwań zasługuje dziś na
największą uwagę i to ono jest szansą na rozwój osobisty każdego człowieka”*

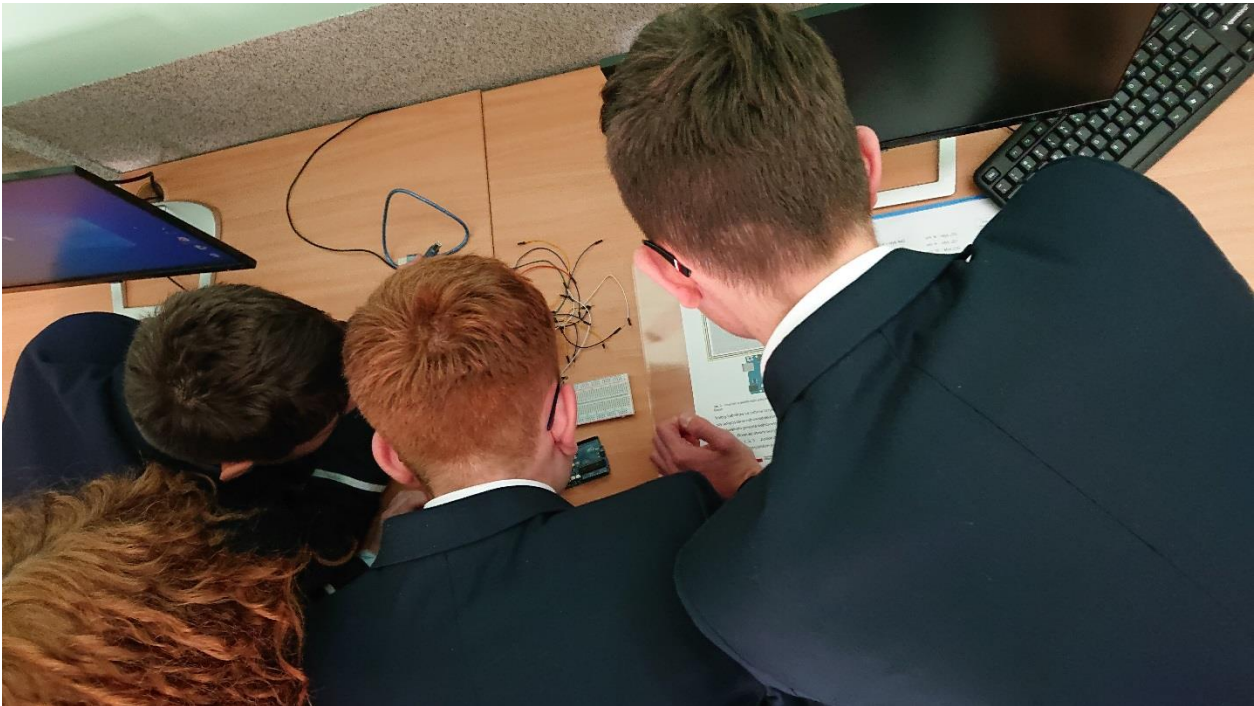
Wojciech Welskop

Drodzy nauczyciele i nauczycielki, oddajemy w Wasze ręce narzędzie, które jak wynika z naszych badań, przełamuje niechęć do nauki szkolnej oraz pozwala dostrzec powiązanie wiedzy zdobywanej w szkole z praktycznym zastosowaniem. Udział w tego typu zajęciach, opartych na interdyscyplinarnym, kreatywnym podejściu do poszukiwania rozwiązań kształtuje kluczowe kompetencje oraz ukazuje powiązanie między różnymi przedmiotami szkolnymi.

Nasza propozycja, to zgodne z podstawą programową kilku przedmiotów, projekty uczniowskie, wspomagane nowatorskimi narzędziami kształtującymi zdolności z zakresu IT i manualne (wręcz inżynierskie), realizowane w ramach obowiązującego planu lekcji, prowadzone przez nauczycieli różnych przedmiotów.

Oferujemy trzy gotowe komplety materiałów zajęć interdyscyplinarnych oraz pełne kody źródłowe. Można je dowolnie modyfikować, aby uzyskać

pożądany efekt łącząc inne miejsca (piny) lub uzyskać inny napis na ekranie po poprawnym połączeniu kolejnej pary miejsc na płytce stykowej. Dodatkowo w niniejszym podręczniku znajdziecie wiele pomysłów na wykorzystanie elementów elektronicznych dołączonych do zestawu podstawowego Arduino Uno. Każdy opis jest bogato zilustrowany, jak należy zmontować układ, aby dostać pożądany efekt.

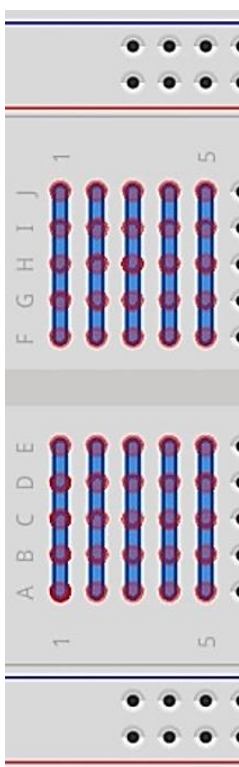


Rysunek 1 – uczniowie w trakcie pracy na zajęciach

2 Wskazówki dotyczące modyfikacji zajęć oraz kodów źródłowych

2.1 Temat: kolonializm i jego upadek

Kod źródłowy oparty jest na odnajdywaniu podczas zajęć czterech par punktów na mapie (płytkce stykowej) będących rozwiązaniami kolejnych zagadek. Ich połączenie odbywa się pośrednio poprzez płytkę stykową. Współrzędne pionowe styków *B*, *C*, *D* i *E* na płytce są wewnątrz połączone ze współrzędną *A*, zaś *F*, *G*, *H* oraz *I* z *J* (rys. 2).



Innymi słowy, kod źródłowy rozpozna każdy z pięciu styków jako ten sam pin.

Zatem wystarczy, aby punkty te z linijki o współrzędnej *A* lub *J* (poziomo) powinny zostać połączone kabelkami z pinami mikrokontrolera Arduino w kolejności:

zagadka 1 – piny 7 i 8

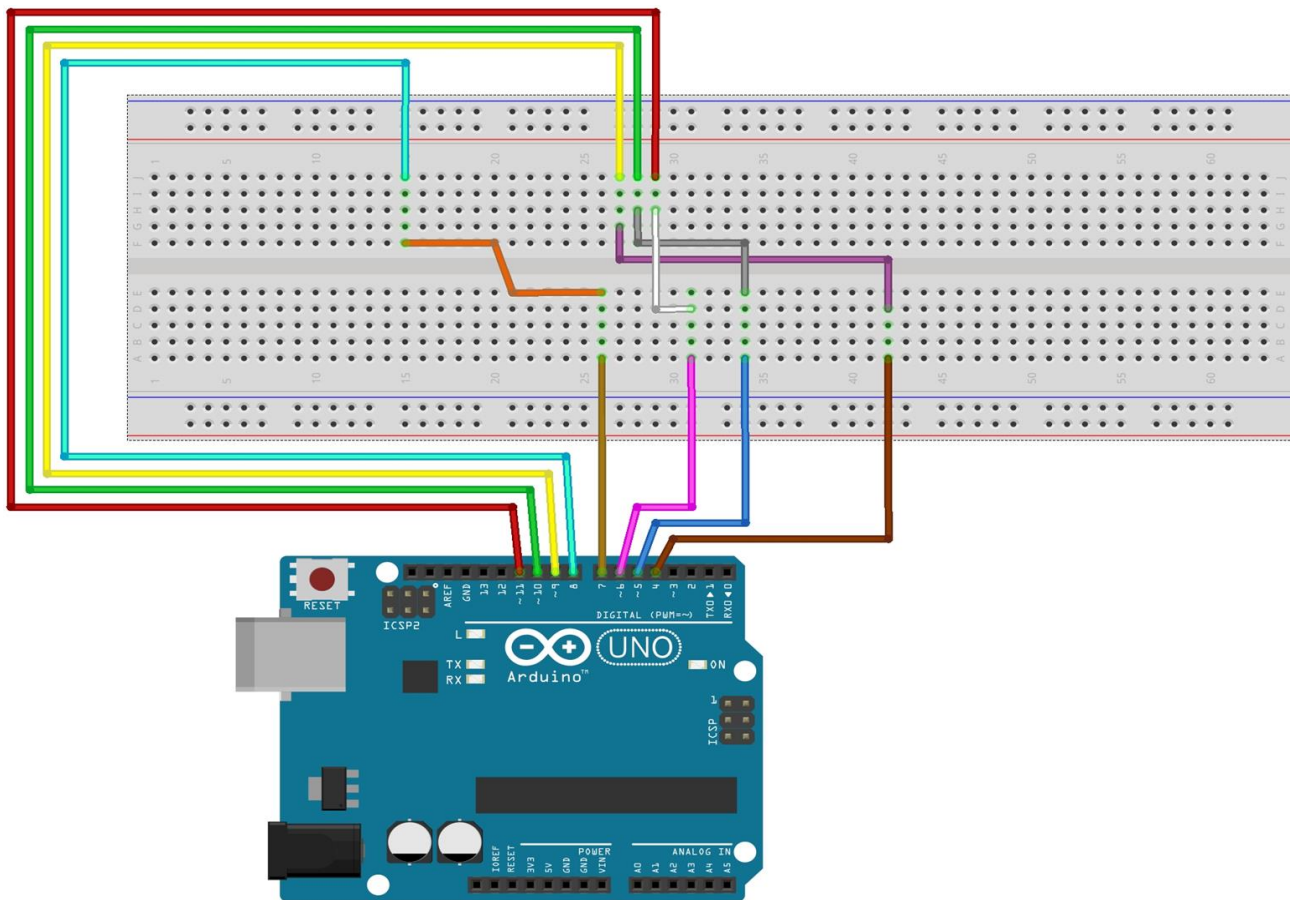
zagadka 2 – piny 4 i 9

zagadka 3 – piny 5 i 10

zagadka 4 – piny 6 i 11.

Rysunek 2 – Punkty płytki stykowej, które są wewnątrz z sobą połączone

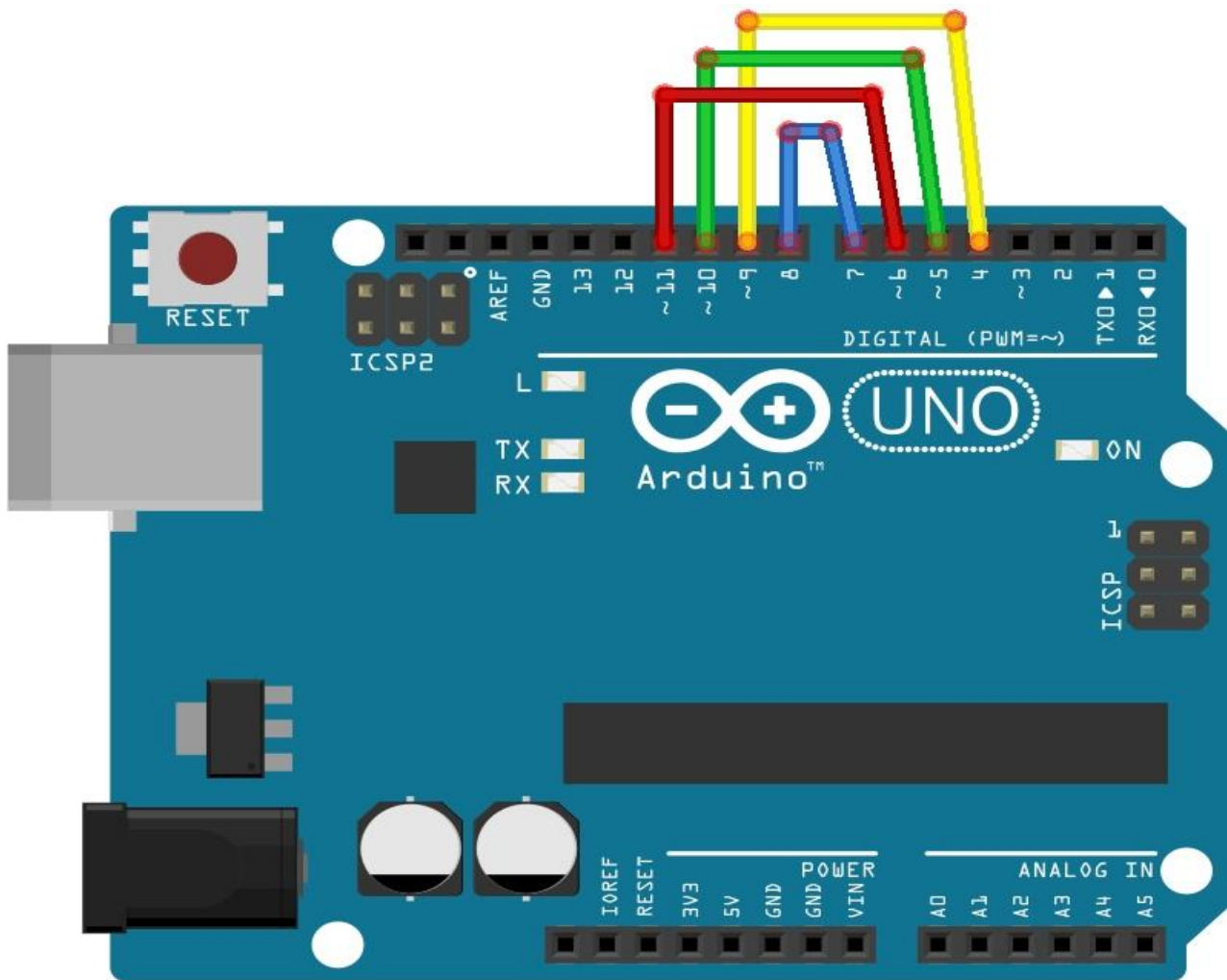
Gotowy układ połączeń po wykonaniu wszystkich zadań będzie wyglądał jak na rysunku poniżej:



fritzing

Rysunek 3 – Gotowy układ po zakończeniu ćwiczenia

Połączenia prowadzące przez płytkę stykową powodują w zasadzie połączenia odpowiednich styków mikrokontrolera. Program można również testować łącząc bezpośrednio odpowiednie styki w odpowiedniej kolejności – da to ten sam efekt, jak połączenia przez płytkę. Można się w ten sposób upewnić, że program działa prawidłowo (rys. 3).



Rysunek 4 – Krótkie połączenie odpowiednich styków bez pośrednictwa płytki stykowej, przydatne do testowania programu

Zademonstrowany program można zmodyfikować, aby sprawdzał połączenia pomiędzy dowolnymi parami styków. Należy wówczas połączyć kontroler Arduino UNO z płytką stykową tak, aby uwzględnić wybrane przez autora zajęć punkty na mapie. Wybrana przez nas mapa może być również podmieniona na dowolną inną ilustrację. Komunikaty, żeby nie znajdowały się bezpośrednio w kodzie programu, zostały zakodowane wartościami z tablicy znaków ASCII (rysunek nr 4). Sposób kodowania można oczywiście dodatkowo skomplikować, aby nie było takie łatwe ręczne odkodowanie treści komunikatów.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Źródło: LookupTables.com

Rysunek 5 – Tabela podstawowych znaków ASCII

2.1.1. Wskazówki modyfikacji kodu źródłowego

Bezpośrednio po uruchomieniu kodu źródłowego, w dedykowanym oprogramowaniu można wprowadzać modyfikacje. Najprostsze z nich wypunktowaliśmy poniżej.

Bardziej skomplikowane modyfikacje programu wymagają elementarnej znajomości podstaw języka C++, na którym jest oparty język programowania Arduino.

```
bool connection78 = false;
bool connection49 = false;
bool connection510 = false;
bool connection611 = false;
```

zmienne zapamiętujące, czy wykonano kolejne części zadania, czyli czy połączone zostały piny 7 i 8, 4 i 9, 5 i 10 oraz 6 i 11

Nauczycielom rozpoczynającym przygodę z mikrokontrolerami proponujemy nie zmieniać tych zapisów, lecz punkty na mapie dopasować do tych ustawień.

zaszyfrowane komunikaty, wyświetlające się w okienku monitora portu szeregowego zapisane w kodzie ASCII (decyduje o tym 0 na końcu ciągu znaków)

```
char s1[5] = { 49, 56, 54, 48, 0 };
char s2[5] = { 49, 52, 57, 56, 0 };
char s3[5] = { 49, 56, 54, 57, 0 };
char s4[5] = { 49, 57, 54, 48, 0 };
```

```
char s5[] = ""You found all the points on the world map. \
wait to reveal a message to humanity:";
char s6[102] = {84,104,101,32,109,97,105,110,...};
```

W tym miejscu należy wpisać własne komunikaty (kody), które mają się wyświetlać po poprawnym połączeniu każdej pary oraz po skompletowaniu całości. Istotne dla nas są ich nazwy: s1, s2, s3, s4 i s5.

W zajęciach założono konieczność odgadywania par na mapie w poprawnej kolejności. Dla przykładu jako pierwsze powinny być zetknięte styki 7 i 8, a pozostałe jeszcze nie:

```

if ( !checkConnection( 4, 9 ) &&
    !checkConnection( 5, 10 ) &&
    !checkConnection( 6, 11 ) &&
    !connection78 && checkConnection( 7, 8 )) {
connection78 = true
Serial.println( s1 );
}

```

wykrzyknik przed zmienną oznacza, że wymagany jest brak połączenia, gdy połączenie między 7 i 8 jest aktywne, na ekranie pojawi się napis nazwany wcześniej s1

Podobny zabieg powtórzony jest w kodzie źródłowym przy kolejnych punktach. Naszym zdaniem eliminuje to możliwość przypadkowego sprawdzania poprawnych połączeń. Jeśli jednak prowadzący chciałby dopuścić możliwość odgadywania par bez względu na kolejność, należy usunąć części kodu dotyczące wymagania braku połączeń.

```

if ( connection78 && connection49 &&
    connection510 && connection611 && !printed ) {

```

```

Serial.println( s5 );

```

```

...

```

jeśli wszystkie zadania zostały wykonane, końcowy komunikat (zakodowany przez nas jako s5)

2.1.2. Modyfikacje programu, potrzebne do dodania dodatkowego (piątego) połączenia pinów mikrokontrolera

W miejsce **X** i **Y** należy wstawić numery pinów odczytane z płytki Arduino, które będą stanowiły łączoną dodatkową parę.

Na początku programu należy dodać zmienną:

```
bool connectionXY = false
```

Trzeba również dodać łańcuch znaków, który zostanie wypisany po połączeniu dodatkowej pary pinów np.:

```
char s15[5] = { 49, 56, 54, 48, 0 };
```

Należy przy tym pamiętać, że łańcuch znaków musi się kończyć znakiem o wartości **0**. W tym przypadku znaki są zakodowane kodem ASCII.

Kolejne warunki wyglądają analogicznie jak dla czterech par:

```
if ( !checkConnection( 4, 9 ) &&  
      !checkConnection( 5, 10 ) &&  
      !checkConnection( 6, 11 ) &&  
      !checkConnection( X, Y ) &&  
      !connection78 && checkConnection( 7, 8 ) ) {
```

itd., a dodatkowy warunek będzie miał postać:

```

if ( !connectionXY && checkConnection( X, Y ) &&
    checkConnection( 6, 11 ) &&
    checkConnection( 7, 8 ) &&
    checkConnection( 4, 9 ) &&
    checkConnection( 5, 10 ) ) {
    connectionXY = true;
    Serial.println( s15 );
}

```

Ostatecznie hasło końcowe wyświetli się, gdy zrealizowane zostaną wszystkie połączenia:

```

if ( connection78 && connection49 &&
    connection510 && connection611 && connectionXY &&
    !printed ) {

```

UWAGA

Scenariusz zajęć oraz instrukcje zawierają szczegółowy opis dotyczący najprostszej wersji, gdzie z powodzeniem sprawdza się Arduino UNO.

W przypadku, gdy nauczyciel (sam, lub z pomocą uczniów) zdecyduje się na urozmaicenie szaty graficznej lub rozbudowanie programu o dodatkowe elementy, powinien skorzystać z mikrokontrolera Raspberry Pi ze względu na dużo wyższe parametry. Rekomendujemy Raspberry Pi Pico.

2.2 Temat: rozpady promieniotwórcze

Zajęcia opierają się na odszukaniu i połączeniu początkowego i końcowego izotopu szeregu promieniotwórczego.

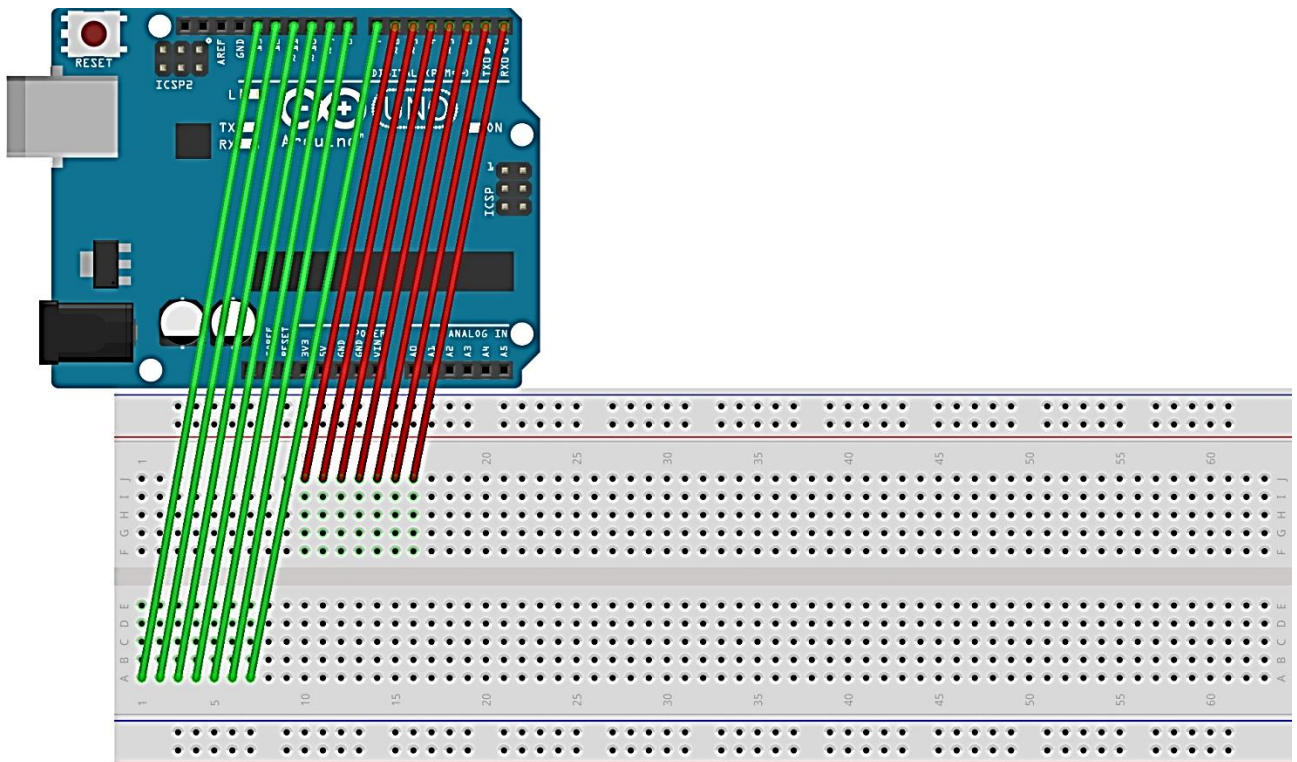
Ta lekcja również może być dowolnie modyfikowana przez nauczycieli, np. poprzez zmniejszenie, zwiększenie lub zmianę zadań, które uczniowie mają wykonać. Można również wykorzystać jedynie pomysł przeniesienia układu okresowego pierwiastków na współrzędne płytki stykowej połączonej z mikrokontrolerem i zbudować dowolną lekcję dotyczącą pierwiastków chemicznych.



Tabela 1 – współrzędne płytki stykowej odpowiadające pinom mikrokontrolera

Pomysł, aby wstępnie zostały połączone aż w czternastu punktach (rysunek 6) sprowadził modyfikację poprawnego rozwiązania zagadek do podmiany jedynie tych dwóch liczb na numery styków odpowiadające poszukiwanym punktom zgodnie z tabelą widoczną obok.

Styki	Pin
F16, G16, H16, I16	0
F15, G15, H15, I15	1
F14, G14, H14, I14	2
F13, G13, H13, I13	3
F12, G12, H12, I12	4
F11, G11, H11, I11	5
F10, G10, H10, I10	6
B7, C7, D7, E7	7
B6, C6, D6, E6	8
B5, C5, D5, E5	9
B4, C4, D4, E4	10
B3, C3, D3, E3	11
B2, C2, D2, E2	12
B1, C1, D1, E1	13



fritzing

Rysunek 6 – schemat prawidłowego połączenia płytki z kontrolerem przed przystąpieniem do rozwiązania zagadki

pięć napisów,
zakodowanych kodem
ASCII; w programie
założono, że napisy
wyświetlą się losowo po
poprawnym połączeniu
dwóch styków

```
char s0[206] = {87, 122, 114, 111, ..., 0};
char s1[175] = {80, 114, 122, 101, ..., 0};
char s2[223] = {82, 101, 103, 117, ..., 0};
char s3[171] = {83, 116, 97, 108, ..., 0};
char s4[160] = {51, 48, 32, 103, ..., 0};
```

2.3 Temat: masa powietrza w sali lekcyjnej

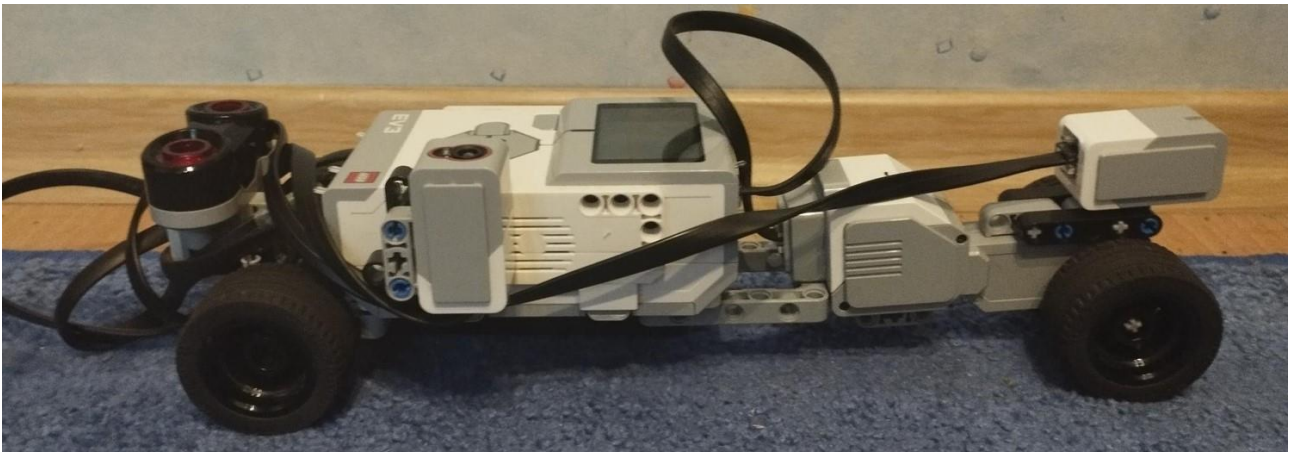
W trakcie zajęć uczniowie muszą wykorzystać mikrokontroler będący częścią zestawu edukacyjnego klocków (np. rekomendowany przez nas Mindstorms® EV3) do wyznaczenia objętości obliczenia masy i oszacowania liczby osób, które mogą przebywać w stworzonej przez nich bazie kosmicznej.

Techniczny charakter zajęć to wymuszająca inżynierską kreatywność budowa „łazika” z klocków Lego®. Jak się okazało, nie stanowi to żadnej przeszkody nawet dla najmłodszych uczniów. W razie problemów uczniowie mogą skorzystać z zestawów podpowiedzi: 💡 (zdjęć gotowego pojazdu) oraz 🧩 (zdjęć przypominających instrukcję budowy). W ostateczności polecamy prowadzącym zajęcia udostępnić uczniom pełną instrukcję budowy najprostszego robota (np. dołączoną do zestawu).



Rysunek 7 – autorskie pomysły uczniów na „łaziki” Lego®.

Pojazdy powstałe podczas zajęć

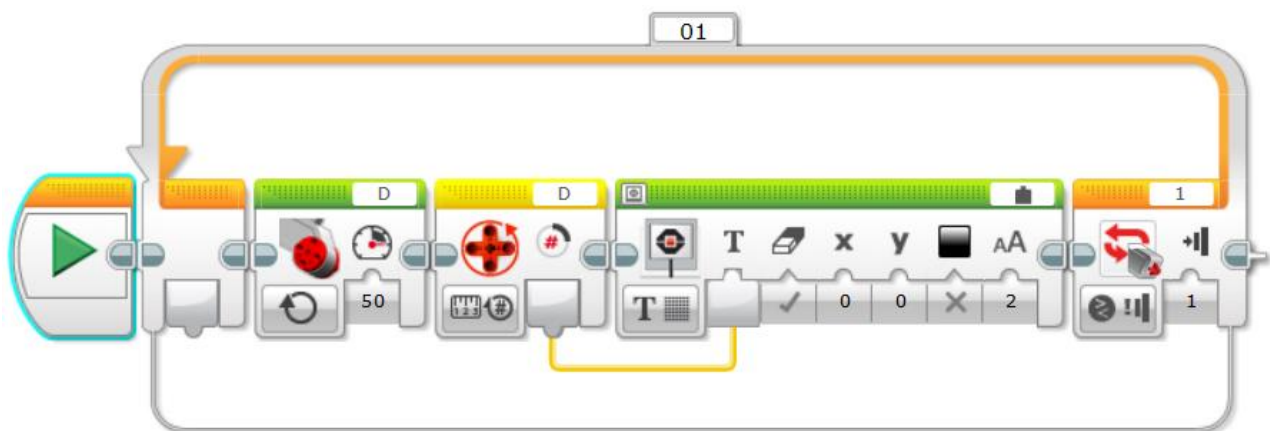


Rysunek 8 – autorskie pomysły uczniów na „łaziki” Lego®

Zauważalne są nie tylko pomysłowość uczniów, ale również spełnienie wymogu użycia minimalnej liczby klocków.

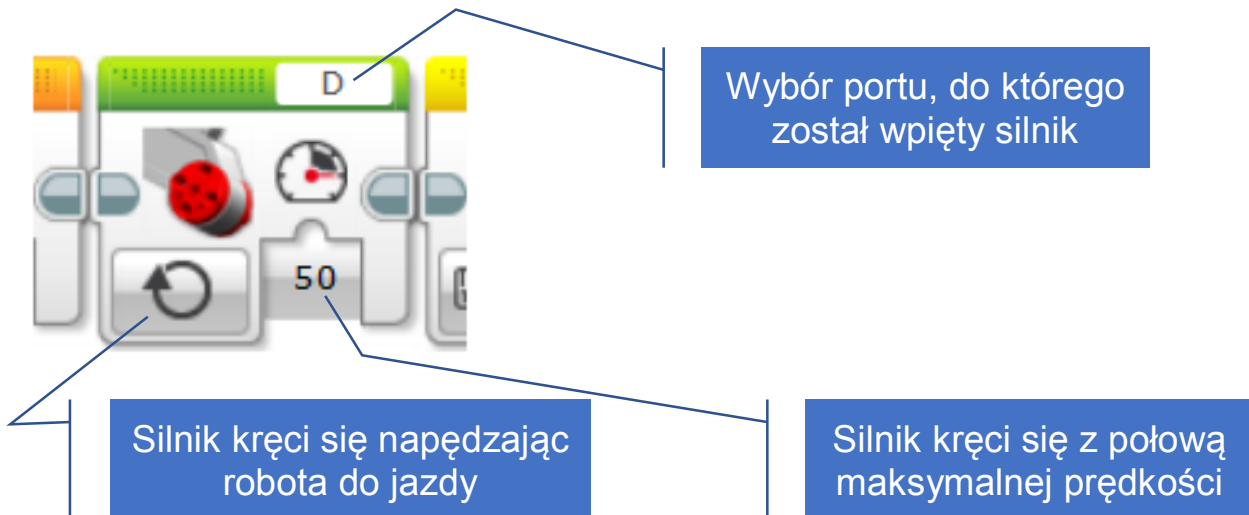
Problem może pojawić się podczas próby wykorzystania oprogramowania do sterowania robotem. Uczniowie ponownie mogą skorzystać z dwóch rodzajów podpowiedzi: naprowadzających 💡 oraz proponujących gotowe rozwiązania 🧩. W ostateczności prowadzący zajęcia może udostępnić uczniom gotowe kody dołączone do materiałów.

Plik „obroty” to prosty program wyświetlający na ekranie kontrolera liczbę obrotów silnika:



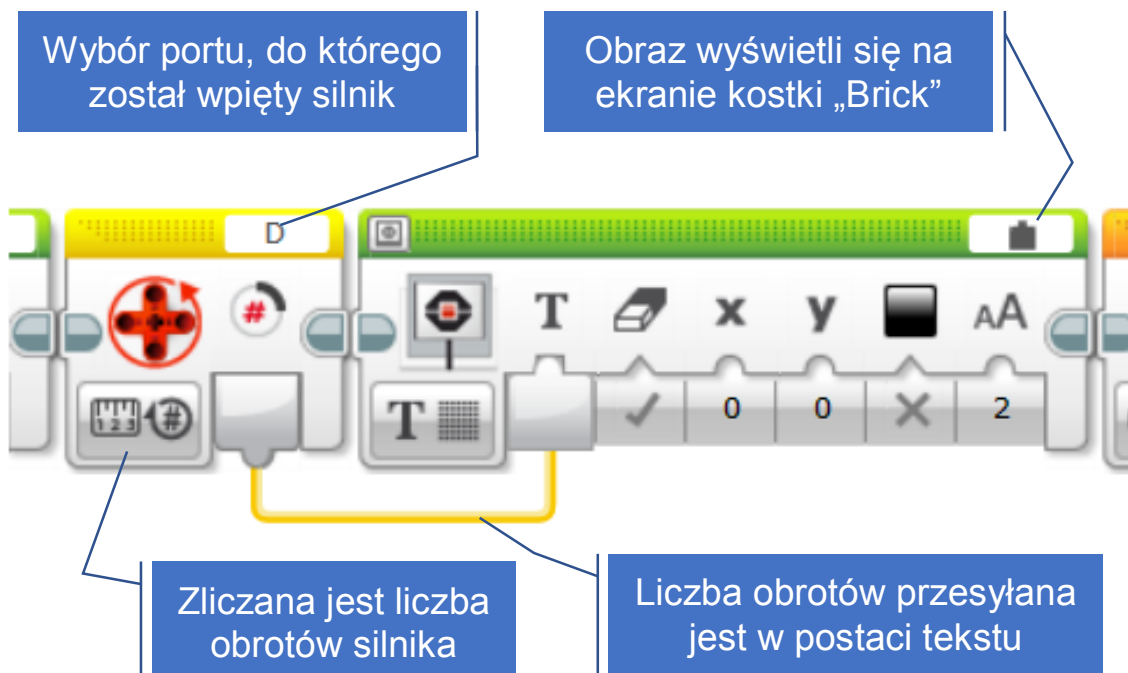
Rysunek 9 – Kod programu zliczającego obroty kół

Pierwszy bloczek to sterowanie silnikiem napędzającym robota.



Rysunek 10 – bloczek sterowania silnikiem

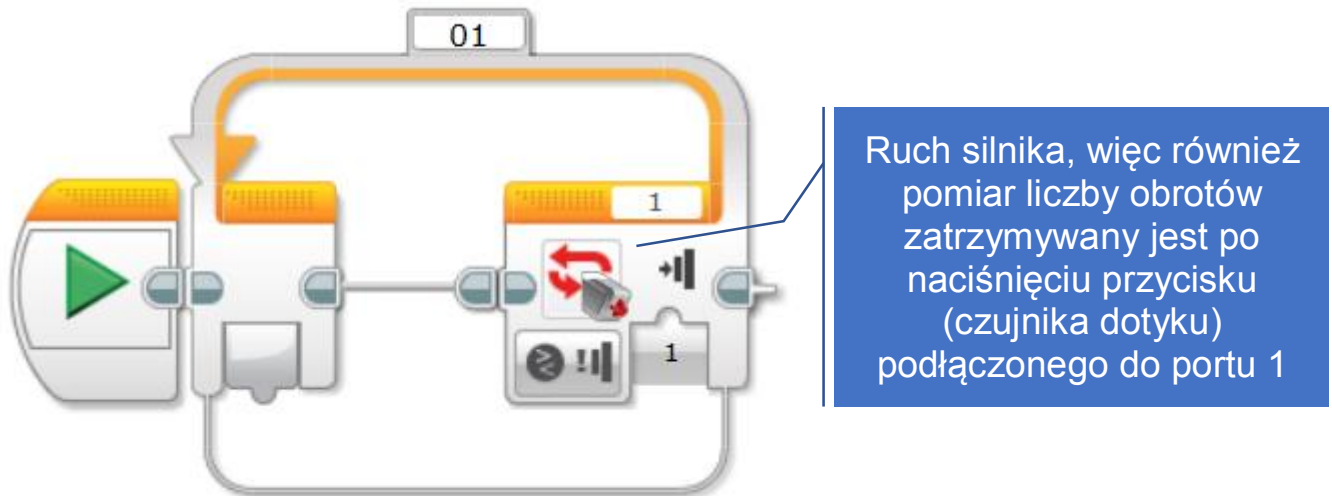
Program zlicza liczbę obrotów i wysyła wynik na ekran.



Rysunek 11 – bloczki licznika i wyświetlacza

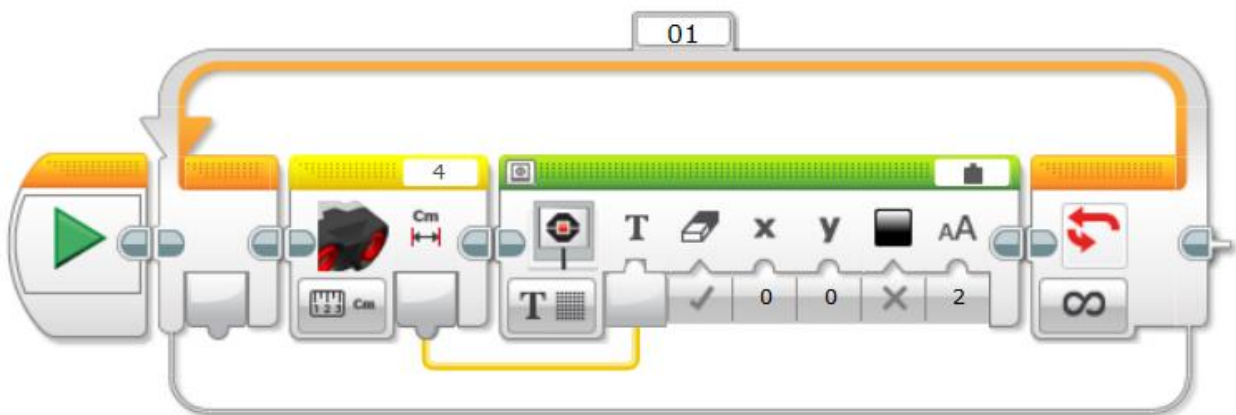
Pozostałe ustawienia ekranu (zielony bloczek) to położenie, kolor i rozmiar tekstu.

Całość zamknięta jest w powtarzalnej w nieskończoność pętli.



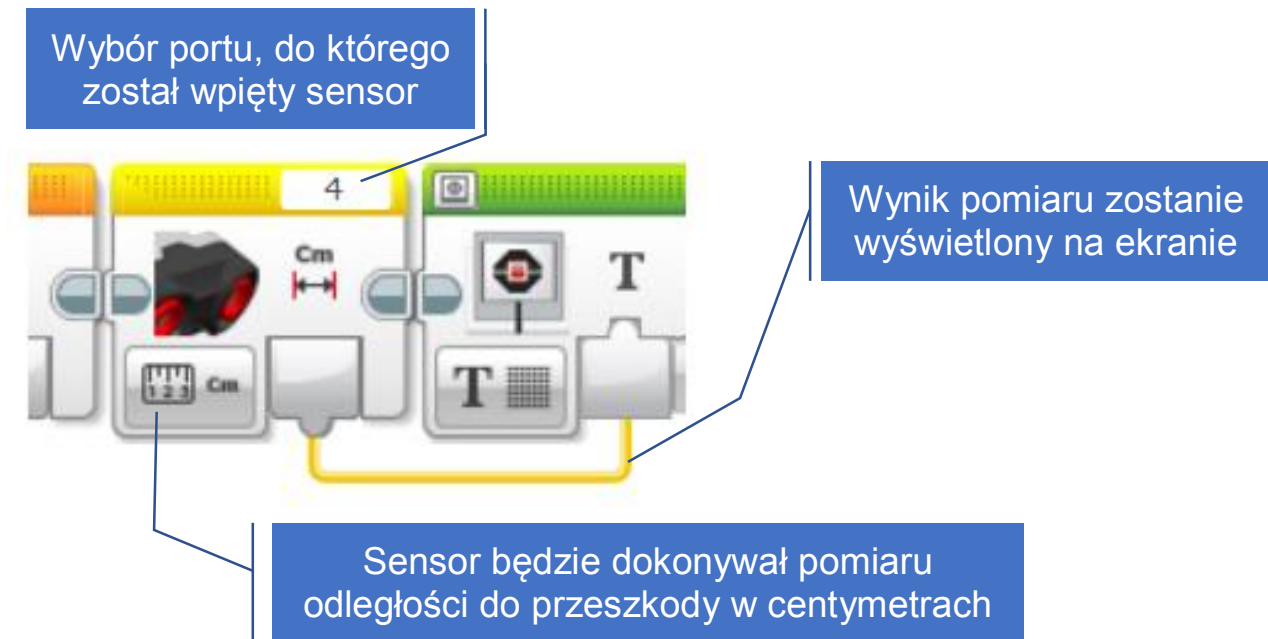
Rysunek 12 – bloczek pętli

Plik „odleglosc” działa w podobny sposób:



Rysunek 13 – Kod programu mierzącego odległość do przeszkody

Bieżący wynik pomiaru odległości będzie wyświetlany dopóki nie zostanie zatrzymany program.



Rysunek 14 – Bloczek sensora odległości

Ponieważ programy są intuicyjne i nie wymagają znajomości języka programowania, uczniowie rzadko wymagają pomocy przy ich tworzeniu. Polecamy urozmaicenie zadań, z wykorzystaniem pozostałych czujników, np. rozpoznającego kolor, oświetlenie, odbicie światła. Można również zaprogramować zbieranie informacji i dokonywanie bieżących obliczeń.

3 Opis dodatkowych 5 elementów uatrakcyjnających zajęcia na przykładzie tematu Kolonializm i jego upadek

3.1. Wyświetlacz LCD

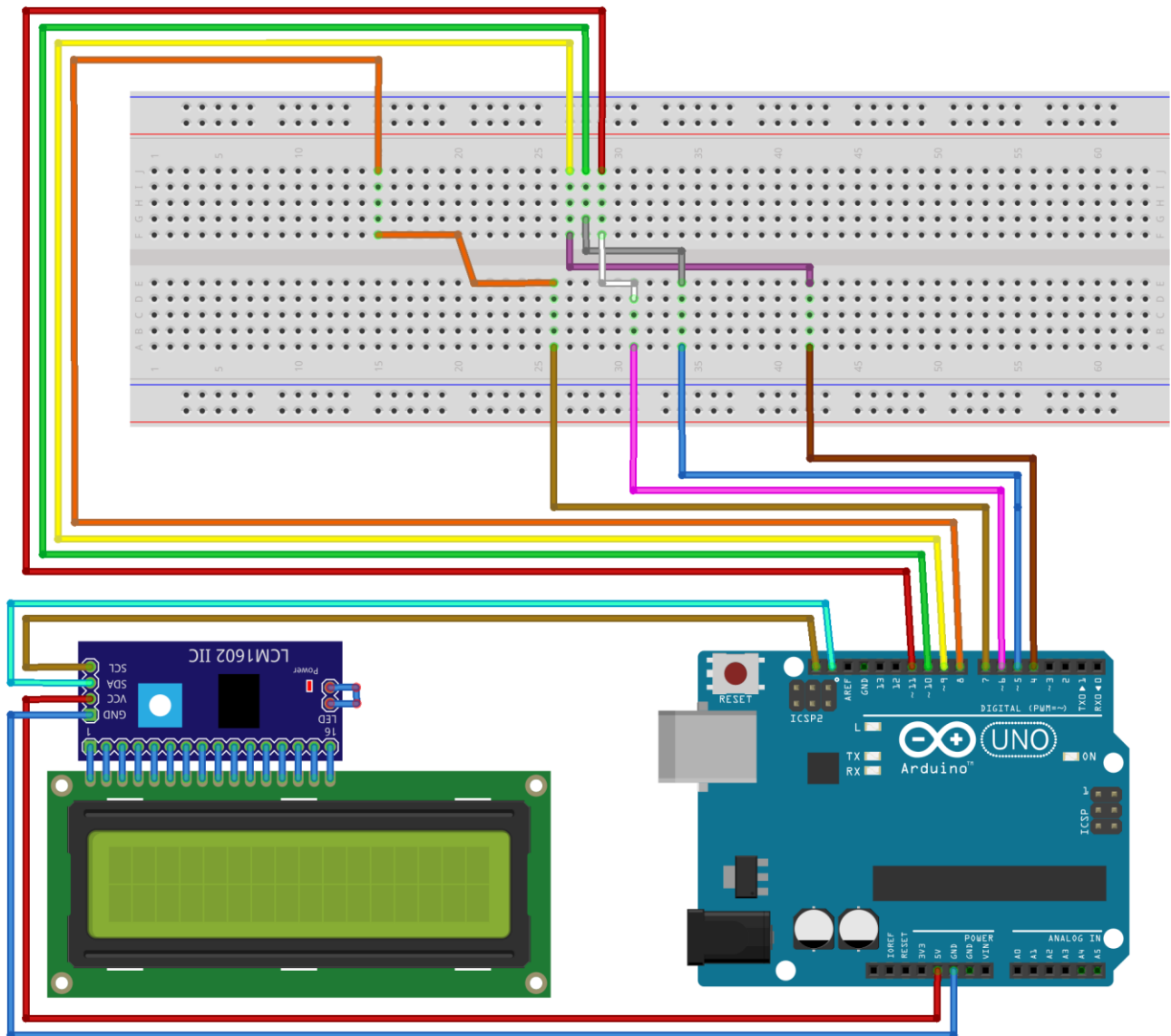
Do wyświetlania informacji można zamiast monitora portu szeregowego wykorzystać wyświetlacz LCD. Najczęściej w zestawach Arduino dostępne są wyświetlacze, które posiadają po 16 znaków w dwóch liniach.

Można sterować wyświetlaczem LCD bezpośrednio z pinów mikrokontrolera, jednakże jest to niewygodne ze względu na fakt, iż wykorzystalibyśmy wówczas większość pinów Arduino. Znacznie wygodniej jest sterować wyświetlaczem za pomocą dodatkowego elementu za pośrednictwem magistrali I²C, jakim jest moduł LCM1602. Często w zestawach edukacyjnych Arduino moduł I²C jest przylutowany do wyświetlacza, wtedy możemy sterować wyświetlaczem tylko za pośrednictwem magistrali I²C.

Magistrala I²C jest specyficzną odmianą sieci komputerowej, przeznaczoną do komunikacji pomiędzy mikrokontrolerami i różnego rodzaju czujnikami lub elementami wykonawczymi. Wykorzystuje dwa przewody (SDA do przesyłania danych i SCL do synchronizacji transmisji). Arduino posiada dwa piny (16 i 17) dedykowane tylko do obsługi magistrali I²C. Aby połączyć mikrokontroler z modułem I²C, należy połączyć ze sobą piny SDA (Arduino i modułu) oraz osobno SCL (również Arduino i modułu), jak na załączonym schemacie. Do prawidłowego działania modułu i wyświetlacza niezbędne jest również podłączenie zasilania +5V do pinu VCC modułu LCM1602 oraz masy (GND). To połączenie również jest widoczne na schemacie.

Moduł LCM1602 posiada również potencjometr, który służy do regulacji jasności wyświetlacza. Efektem źle wyregulowanej jasności wyświetlacza będzie stan, gdy wyświetlacz będzie wygaszony lub zaświeci się całkowicie (wszystkie punkty), w konsekwencji obu przypadków odczyt tekstu będzie niemożliwy.

Do obsługi wyświetlacza posłużą nam biblioteka Liquid Crystal I2C, jeśli używamy jej po raz pierwszy, może wystąpić konieczność doinstalowania jej za pomocą Menadżera Bibliotek, znajdującego się w menu Narzędzia → Zarządzaj Bibliotekami głównego okienka Arduino IDE.



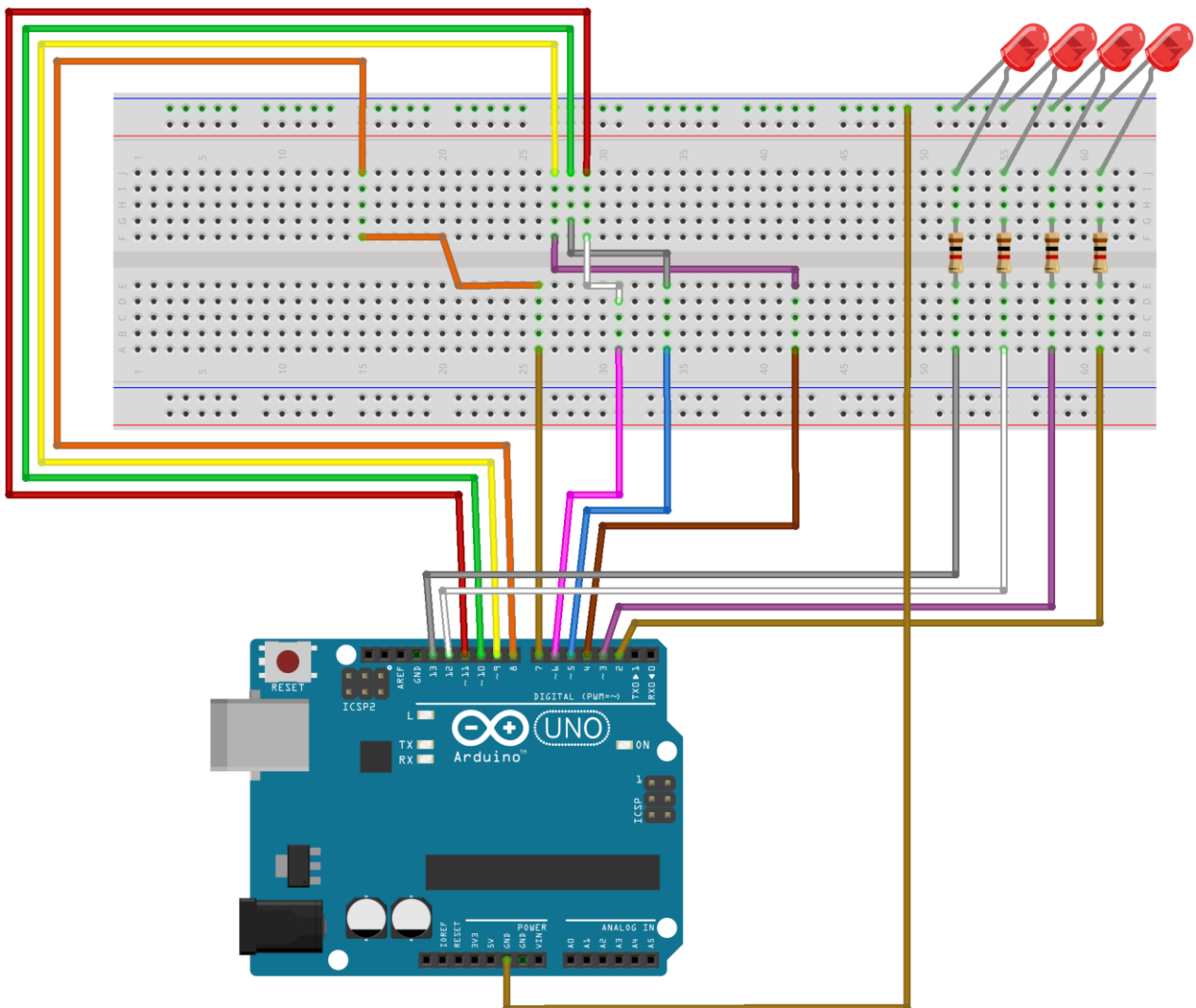
fritzing

Rysunek 15 – przykładowe podłączenie płytki stykowej i wyświetlacza LCD do zajęć 1

Do podręcznika dołączony został kod źródłowy „geo_history_lcd.ino”, który realizuje identyczne zadania, lecz z wykorzystaniem dołączonego wyświetlacza LCD.

3.2. Diody LED

Najprostszym rodzajem komunikacji pomiędzy mikrokontrolerem a człowiekiem jest zaświecająca się w odpowiednich momentach dioda LED. W tym przykładzie wykorzystaliśmy odpowiednie połączenie dodatkowych czterech pinów Arduino (2, 3, 12, 13) z diodami poprzez rezystory 1k Ω , jak na dołączonym schemacie. Rezystory są niezbędne, żeby ograniczyć prąd diody i zabezpieczyć wyjście Arduino przed uszkodzeniem. Anoda diody (dłuższa nóżka) musi być połączona z rezystorem, katoda (krótsza) podłączona do masy układu (GND).



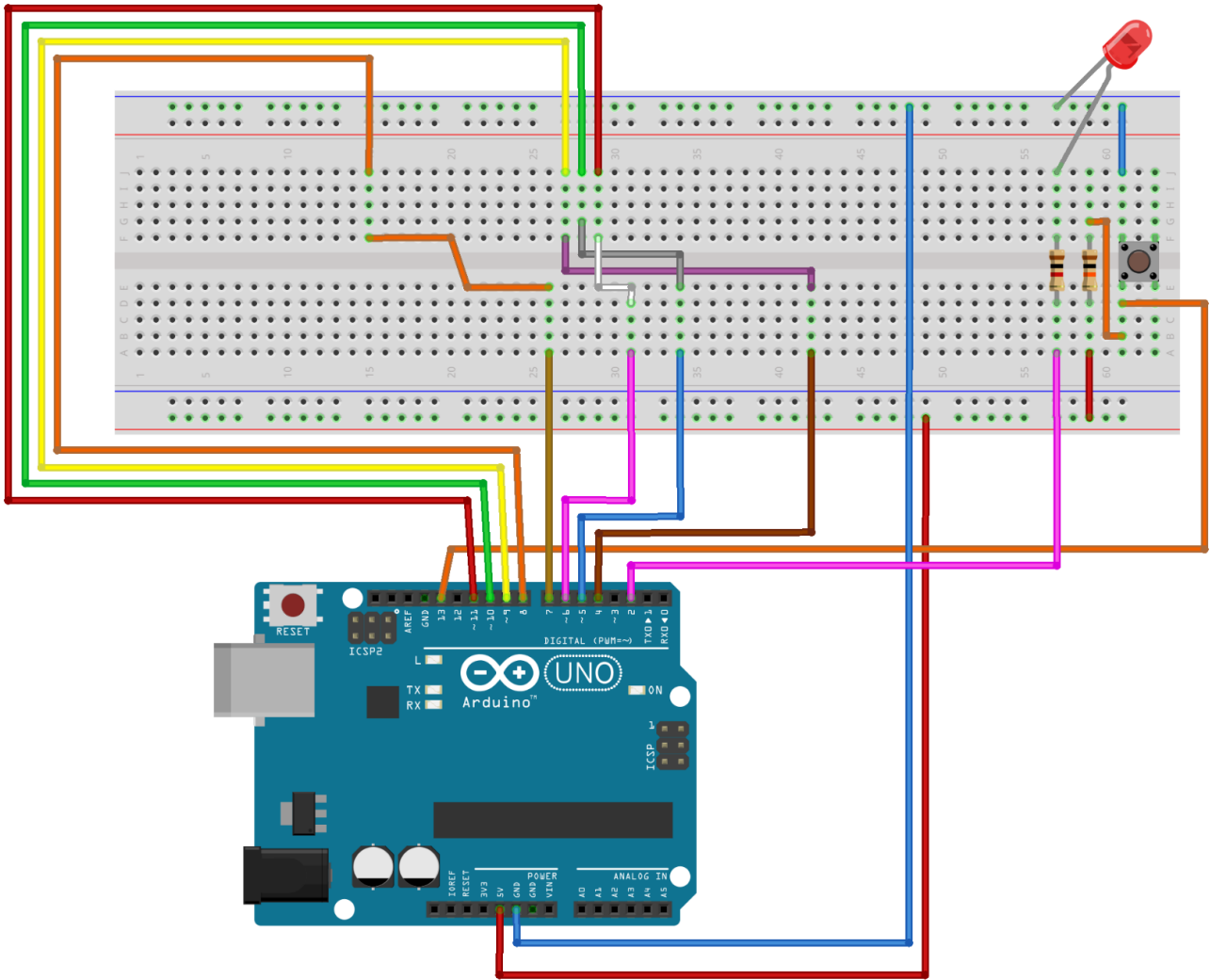
fritzing

Rysunek 16 – przykładowe podłączenie płytki stykowej i diod LED do zajęć 1

3.3. Przycisk

W tym przykładzie wykorzystano przycisk oraz diodę LED. Po poprawnym połączeniu dwóch miejsc na mapie zaświeca się dioda. Musimy przycisnąć przycisk, aby wyświetliła się odpowiednia data, wtedy dioda gaśnie. Sposób połączenia dodatkowych elementów przedstawia dołączony schemat.

W celu wykrywania, czy przycisk jest wciśnięty czy nie, łączymy go z masą układu (GND) i z drugiej strony szeregowo z rezystorem – jeśli wciśniemy przycisk przez ten fragment obwodu popłynie prąd i na odpowiednim wejściu Arduino pojawi się stan niski. Wykrycie tego stanu powoduje wykonanie odpowiednich instrukcji (zgaszenie diody i wyświetlenie odpowiedniej daty). Wartość rezystora nie ma znaczenia, wciśnięty przycisk będzie zawsze miał wielokrotnie mniejszą rezystancję niż dowolny rezystor z tych dostępnych w zestawach Arduino.

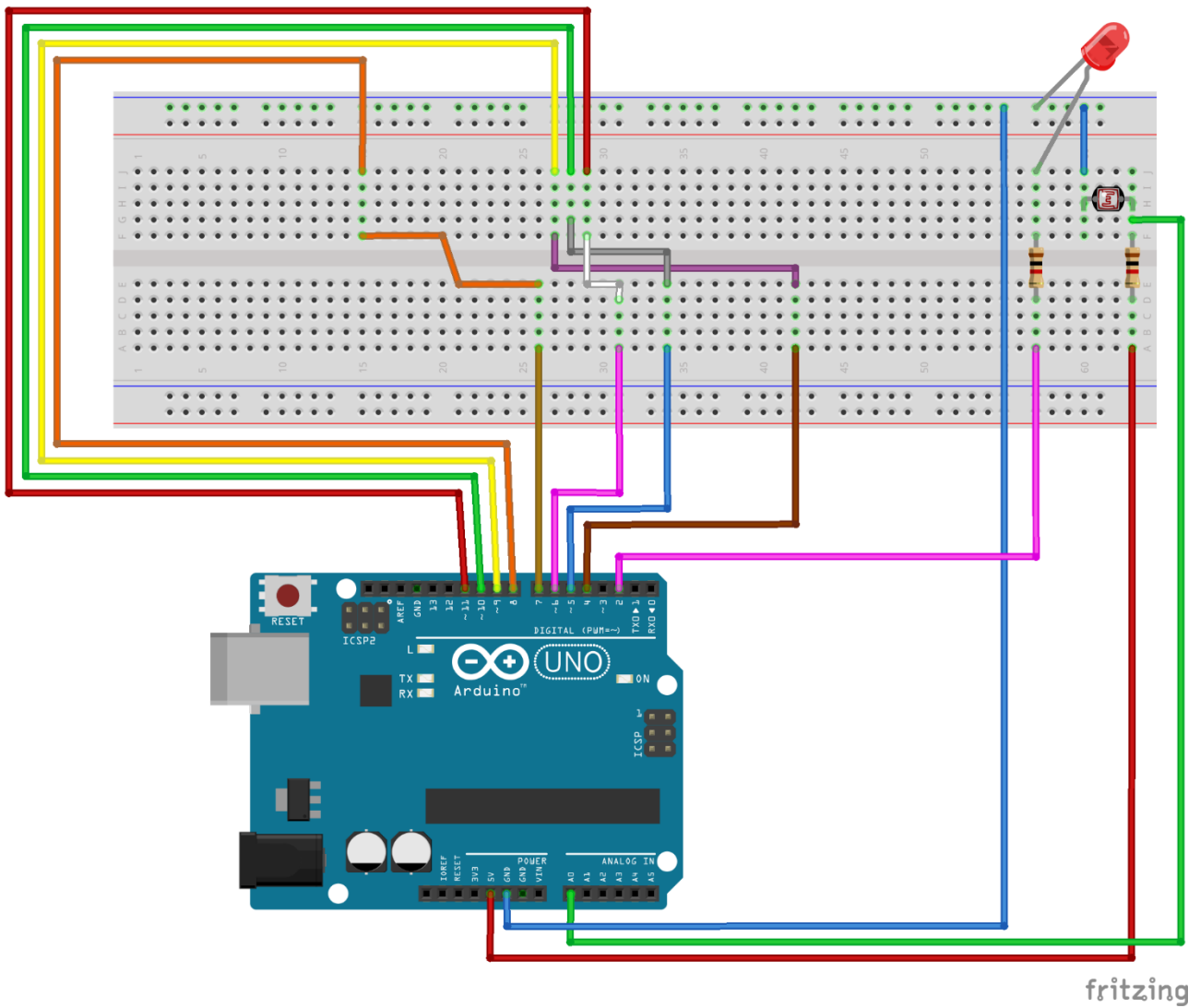


fritzing

Rysunek 17 – przykładowe podłączenie płytki stykowej, przycisku i diody LED do zajęć 1

3.4. Fotorezystor

Fotorezystora można użyć do sterowania układem podobnie jak przycisku, z tą różnicą, że do odczytu wartości napięcia należy zastosować któreś z wejść analogowych układu Arduino (A0..A5), np. pinu A0. W odróżnieniu od pinów cyfrowych piny A0 do A5 są tylko wejściami, co powoduje, że nie trzeba ich w żaden sposób konfigurować, działają od razu po włączeniu zasilania. Zastąpienie fotorezystora powoduje zmniejszenie przepływu prądu przez odpowiedni fragment układu, co powoduje odczyt przez Arduino większej wartości z pinu A0. Piny od A0 do A5 są wejściami analogowymi. Tak naprawdę mikrokontroler umieszczony na płycie Arduino mierzy wartość napięcia na odpowiednim pinie analogowym, zakładając, że zawiera się ono pomiędzy 0V (czyli masą układu, GND) a napięciem odniesienia (pin IOREf). Rezystancja fotorezystora zależy od oświetlenia, im jaśniej tym jest mniejsza, im ciemniej tym bardziej rośnie.

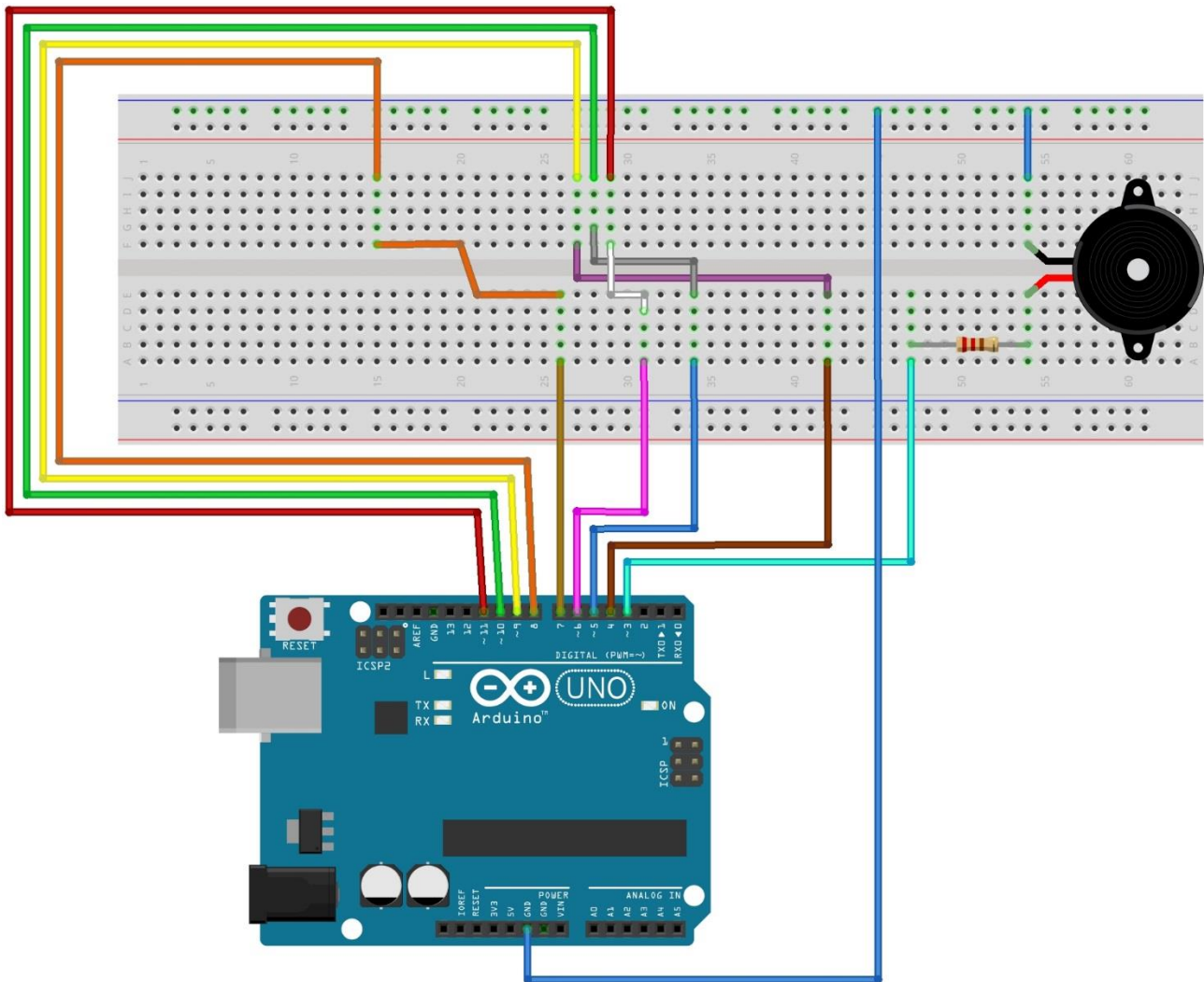


Rysunek 18 – przykładowe podłączenie płytki stykowej, fotorezystora i diody LED do zajęć 1

3.5. Buzzer

W tym przykładzie do dodatkowej sygnalizacji prawidłowej odpowiedzi wykorzystano głośniczek piezoelektryczny, dołączony do zestawu.

Prawidłowe połączenie dwóch punktów na mapie powoduje wydanie dźwięku o częstotliwości 1000Hz przez 1s. W tym celu należy połączyć dodatkowy obwód jak na dołączonym schemacie.

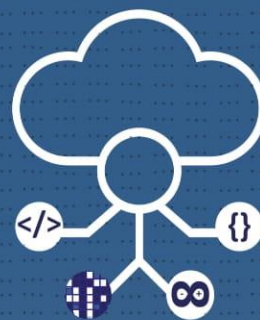


fritzing

Rysunek 19 – przykładowe podłączenie płytki stykowej i głośniczka do zajęć 1

3.6. Łączenie poszczególnych wariantów

Możliwe jest oczywiście łączenie poszczególnych wariantów. Należy tylko zadbać o odpowiednią konfigurację i połączenie wyjść i wejść. Należy również scalić odpowiednie fragmenty kodów programów w jeden, obsługujący na przykład wyświetlacz LCD, fotorezystor, diodę LED i głośniczek. W zestawie znajduje się wiele innych czujników i elementów wykonawczych, które można z powodzeniem wykorzystać do interaktywnej obsługi mikrokontrolera, np. wielokolorowa dioda LED, pilot z czujnikiem podczerwieni, czujnik hałasu, wyświetlacze siedmiosegmentowe i wiele innych.



Mikrokontrolery MEGAprojekty

Dariusz Świder i **Damian Kimla** to nauczyciele, którzy nieszablonowo podchodzą do kwestii nauczanych przez siebie przedmiotów.

Są pasjonatami i swą pasją zarażają uczniów.

Zaprezentowane przez nich rozwiązanie to połączenie trzech koncepcji: interdyscyplinarność, grupowa praca metodą projektu i eksperyment z wykorzystaniem mikrokontrolerów, klocków i dodatkowych elementów. Projekty pomagają kształtować kompetencje niezbędne we współczesnym świecie: twórcze i kreatywne podejście do rozwiązywania problemów, umiejętność pracy i współpracy w grupie oraz rozwijanie zdolności w zakresie ICT.

