

STEMKIT

4SCHOOLS

WYKORZYSTANIE CZUJNIKÓW PODCZERWIENI W SYSTEMACH ALARMOWYCH SCRATCH 2.0 I GPIO

PLAN LEKCJI 1



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

Projekt ten został sfinansowany przy wsparciu Komisji Europejskiej.

Niniejszy komunikat odzwierciedla jedynie poglądy autora, a Komisja nie ponosi odpowiedzialności za jakiegokolwiek wykorzystanie informacji w nim zawartych.



Spis treści

1.	Wykorzystanie czujników podczterwieni w systemach alarmowych.....	2
1.1	Wprowadzenie.....	2
1.1.1	Ogólny opis.....	2
1.1.2	Cele nauczania	2
1.1.3	Odniesienia do programu nauczania.....	3
1.1.4	Spis niezbędnych materiałów	3
1.1.5	Czas trwania	3
1.2	Plan lekcji	4
1.2.1	Wprowadzenie	4
1.2.2	Przygotowanie	4
1.2.3	Wykonanie	9
1.2.4	Wnioski	10
1.2.5	Dodatkowe ćwiczenie (nieobowiązkowe)	10
1.3	Bibliografia i dodatkowe zasoby.....	11



1. Wykorzystanie czujników podczerwieni w systemach alarmowych

1.1 Wprowadzenie

1.1.1 Ogólny opis

Ta lekcja wprowadza użycie odblaskowego czujnika optycznego TCRT5000 z wyjściem tranzystorowym do zaprojektowania prostego układu, który będzie działał jako alarm dla okien otwieranych przez nieupoważnione osoby. Czujnik może zostać umiejscowiony obok ramy okiennej dzięki zakresowi jego działania, który wynosi od 0,2 mm do 15 mm.

1.1.2 Cele nauczania

Główne cele nauczania zawarte w tym planie lekcji to:

- Zrozumienie koncepcji i zawartości sprzętu w celu utworzenia kodu w Scratchu.
- Zapoznanie się z ćwiczeniami związanymi z budowaniem konsoli w celu usprawnienia eksperymentowania z tematami związanymi z STEM.
- Zapoznanie się z odczytami z pinów GPIO.
- Zrozumienie kroków kodowania w Scratchu.
- Projektowanie kodowania w Scratchu.
- Używanie czujników, emitera podczerwieni, fototranzystora i innych elementów do eksperymentowania z konsolą STEMKIT.
- Wykonywanie kodowania na konsoli STEMKIT na przykładzie czujników IR w systemach alarmowych.
- Wykonanie podstawowego montażu czujników na płytce stykowej.
- Eksperymentowanie z podłączeniem czujników do Raspberry Pi i konsoli STEMKIT.
- Autonomia w tworzeniu prostego obwodu, który może służyć jako demonstracja instalacji bezpieczeństwa w budynku.
- Autonomia we wprowadzaniu pojęć kodowania w środowisku klasowym.



1.1.3 Odniesienia do programu nauczania

Domeny, subdomeny, przedmioty/tematy, z którymi można powiązać ten plan lekcji, to:

- Nauka (fizyka/chemia/biologia/geologia): napięcie, moc, obwody, wyzwalacze alarmów, metoda naukowa, badania, eksperymenty, analiza i interpretacja wyników
- Informatyka: procesory i peryferia, interfejsy, język programowania i główne struktury, kodowanie
- Technologia: elektronika, sprzęt i oprogramowanie typu open source, czujniki, sygnał cyfrowy, komputery jednopłytkowe, konsola

1.1.4 Spis niezbędnych materiałów

Do realizacji tej lekcji potrzebna jest konsola STEMKIT z Raspberry Pi wraz z następującymi elementami:

- 3 czujniki TCRT5000
- 1 brzęczyk z generatorem
- 2 przewody połączeniowe żeńsko-żeńskie
- 5 przewodów połączeniowych męsko-żeńskich
- Rezystory 3 x 10k Ω
- Rezystory 3 x 330 Ω
- 1 płytki stykowa

1.1.5 Czas trwania

Czas trwania lekcji szacuje się na około 45-60 minut, tj. 1 godzinę lekcyjną.

1.2 Plan lekcji

Plan lekcji podzielony jest na cztery części: wprowadzenie, przygotowanie, wykonanie i wnioski. Jako uzupełnienie istnieje również nieobowiązkowe do wykonania ćwiczenie znajdujące się na końcu.

1.2.1 Wprowadzenie

TCRT5000 jest czujnikiem, który składa się z dwóch elementów w jednej obudowie. Emiter emituje falę o długości 950 nm, która jest następnie odbierana przez fototranzystor pracujący jako detektor. Im bliżej obiektu znajduje się czujnik, tym większy będzie odczyt napięcia z fototranzystora. Ponieważ Raspberry Pi posiada piny GPIO, odczyt ten może być śledzony w celu sprawdzenia, czy przeszkoda (rama okna) jest blisko czujnika i w rezultacie, czy okno jest zamknięte.

W tej lekcji Scratch zostanie użyty do zademonstrowania przykładowego kodu, który może być użyty do śledzenia tego prostego obwodu.

1.2.2 Przygotowanie

Faza przygotowania wymaga wykonania podstawowego montażu czujników na płytce stykowej i ustawienia kodu w Scratchu. Zaczniemy od płytki.

Umieść trzy czujniki TCRT5000 na pustej płytce stykowej, podłączając emiter fototranzystora i katodę emitera podczerwieni do szyny uziemiającej. Następnie podłącz kolektor fototranzystora za pomocą rezystora 10k Ω do szyny dodatniej na płytce stykowej. Podobnie podłącz anodę emitera podczerwieni do szyny dodatniej za pomocą rezystora 330 Ω . Powtórz to dla pozostałych dwóch czujników TCRT5000.

Teraz czas na podłączenie czujników do Raspberry Pi. Podłącz przewód połączeniowy o odpowiedniej długości do każdego czujnika TCRT5000. Powinien być podłączony do szyny, do której podłączony jest także kolektor fototranzystora i zasilany z szyny napięcia dodatniego za pomocą rezystora 10k Ω . Na tym etapie powinny być trzy przewody połączeniowe – po jednym na każdy czujnik TCRT5000. Przewody te należy podłączyć do pinów GPIO na Raspberry Pi oznaczonych jako 35, 33 i 31 (lub w GPIO: 19, 13 i 6). Teraz nadszedł również czas na podłączenie brzęczyka bezpośrednio do Raspberry Pi. Możesz to zrobić, podłączając pin GND brzęczyka do pinu 39, a jego przewód dodatni do pinu 38 (lub GPIO: 26). Wreszcie płytka stykowa musi otrzymać zasilanie z Raspberry Pi. W tym celu można wykorzystać szynę +5V z Raspberry Pi (pin 4) oraz GND (pin 6). Użyj przewodów połączeniowych, aby zasilić płytkę stykową. Konfiguracja sprzętu jest zakończona, więc teraz możemy przejść do Scratcha.



Wewnątrz Scratcha możesz wybrać dowolne tło, na którym widoczne są co najmniej trzy okna. W tej lekcji użyjemy tła urban1. Ustaw je na całą scenę. Jednocześnie musisz dodać trzy obiekty, które zmieniają swój wygląd po uruchomieniu alarmu. Tutaj sugestią jest użycie duszka, który łączy w sobie button4-a i button5-b. Kod zmieni wygląd na podstawie odczytu z czujników. Ponieważ potrzebujesz trzech duszków (po jednym na każde okno), zduplikuj je, aby uzyskać następującą konfigurację (środkowy duszek ma aktywny drugi wygląd, aby pokazać różnicę):



Obraz 1. Środowisko Scratch z wszystkimi wymaganymi elementami umieszczonymi na ekranie

Źródło: Projekt STEMKIT4Schools

Zanim przejdziemy do głównego kodu, popracujmy nad trzema elementami, które są umieszczone na oknach.

Aby zasymulować alarm, będziemy używali wiadomości nadawanych przez Scratcha, aby odpowiednio reagować na odczyty z czujników. Po przejściu do zakładki Scripts, należy dodać dwie reakcje dla przychodzących wiadomości. Załóżmy, że dla pierwszego czujnika będziemy emitować komunikaty window1-open i window1-closed. Gdy odebraną wiadomością jest window1-open, wtedy musimy uruchomić alarm i zmienić wygląd z zielonego znacznika na czerwony krzyżyk. Podobnie, gdy okno zostanie zamknięte, zielony znacznik powinien pojawić się z powrotem. Poniżej przedstawiony jest przykładowy kod:



Obraz 2. Skrypt dla elementów reagujących na odczyty z pinów GPIO
 Źródło: Projekt STEMKIT4Schools

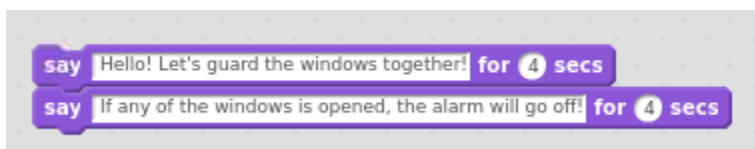
Powtórz tę konfigurację dla pozostałych dwóch elementów umieszczonych nad oknami i pamiętaj o zmianie tytułów komunikatów (window2-open, window2-closed, window3-open, window3-closed).

Główny kod jest miejscem, gdzie zadanie to jest najbardziej interesujące. Po wybraniu kota, przejdź do zakładki Scripts i zacznij dodawać kod. Pierwszą rzeczą jaką chcemy teraz ustawić jest poinformowanie Scratcha, że piny GPIO 19, 13 i 6 powinny być odczytywane jako piny wejściowe.



Obraz 3. Początek kodu - ustawienie odpowiednich pinów GPIO jako wejściowych
 Źródło: Projekt STEMKIT4Schools

Następnie, zanim uruchomimy główny kod, można dodać kilka komunikatów.



Obraz 4. Komunikaty wprowadzające wykonywane przez kod
 Źródło: Projekt STEMKIT4Schools

W kolejnym kroku dodamy pętlę powtórzeń, która wykona 30 cykli, rozumianych jako „sprawdzenia” czujników.



Obraz 5. Pętla powtórzeń

Źródło: Projekt STEMKIT4Schools

W każdej pętli chcemy dotrzeć do wszystkich trzech czujników, aby sprawdzić, czy okna są zamknięte, czy nie. Struktura jest bardzo prosta. Za pomocą instrukcji możemy sprawdzić czy odczyt z odpowiednich pinów GPIO jest wysoki i czy odpowiednio zadziała. Poniżej znajduje się przykład dla pinu GPIO o numerze 19.



Obraz 6. Instrukcja odczytująca wartość GPIO i rozsyłająca komunikaty

Źródło: Projekt STEMKIT4Schools

Zauważ, że jedyną akcją, którą tutaj podejmujemy, jest wyemitowanie wiadomości window1-open lub window1-closed w oparciu o wartość pinu GPIO 19. Podobnie należy dodać bloki dla pozostałych GPIO: 13 i 6. Ponownie, nie zapomnij zmienić numeru pinu GPIO, a także wiadomości, która ma być nadawana.

W kolejnym kroku chcemy również sterować naszym brzęczykiem. Warunkiem jest to, że jeśli wszystkie okna są zamknięte, alarm nie jest aktywny. Gdy co najmniej jedno okno zostanie otwarte, wyemitujemy dźwięk alarmu, a także pozwolimy kotu wypowiedzieć słowo “Alarm!”, aby mieć również uwagę wizualną. W tym celu użyjemy instrukcji z trzema warunkami połączonymi logiką lub instrukcją.



Obraz 7. Deklaracja do wyzwalania alarmu w razie potrzeby
 Źródło: Projekt STEMKIT4Schools

Tuż przed zakończeniem bieżącej pętli, wstrzymamy jej wykonywanie na 1 sekundę. W ten sposób definiujemy częstotliwość wykonywania naszych kontroli, które mają być wykonywane raz na sekundę.



Obraz 8. Wstrzymanie wykonywania kodu na 1 sekundę tuż przed wyjściem z pętli
 Źródło: Projekt STEMKIT4Schools

Na koniec, ponieważ nie chcemy, aby po zakończeniu naszego kodu brzęczyk wydawał dźwięk, zaraz po pętli *repeat* musimy dodać instrukcję ustawiającą pin GPIO numer 26 (ten z brzęczykiem) w stan *low* (czyli bez dźwięku).



Obraz 9. Ustawienie brzęczyka GPIO w stan low po zakończeniu pętli głównej
 Źródło: Projekt STEMKIT4Schools

Końcowy wynik definiujący nasz kod jest przedstawiony poniżej.

```

when clicked
  set qpio 15 to input
  set qpio 13 to input
  set qpio 6 to input
  say Hello! Let's guard the windows together! for 4 secs
  say If any of the windows is opened, the alarm will go off for 4 secs
  repeat 31
    if qpio 15 is high? then
      broadcast window1-open
    else
      broadcast window1-closed
    if qpio 13 is high? then
      broadcast window2-open
    else
      broadcast window2-closed
    if qpio 6 is high? then
      broadcast window3-open
    else
      broadcast window3-closed
    if qpio 15 is high? or qpio 13 is high? or qpio 6 is high? then
      set qpio 24 to output high
      say Alarm!
    else
      set qpio 24 to output low
      say 
    wait 1 secs
  set qpio 24 to output low
  
```

Obraz 10. Główny kod dla przykładowego układu
 Źródło: Projekt STEMKIT4Schools

1.2.3 Wykonanie

Nareszcie możemy uruchomić nasz kod! Wykonaj kolejne podzadania, aby dowiedzieć się więcej o tym przykładzie. Aby uniknąć fałszywych odczytów, upewnij się, że wszystkie elementy użyte do budowy obwodu (zworki lub rezystory) nie zakrywają żadnego z czujników TCRT5000.

Gromadzenie danych

Przy pierwszym uruchomieniu należy upewnić się, że czujniki nie są zasłonięte, czyli innymi słowy, że nie ma żadnych elementów, które mogłyby odbijać emitowaną wiązkę podczerwieni i tym samym wprowadzać piny GPIO w stan *high*. W następnym przebiegu postaraj się zakryć jeden sensor palcem lub innym elementem. Spróbuj zaobserwować zachowanie, gdy wszystkie trzy czujniki są zasłonięte, symulując sytuację, w której wszystkie okna są zamknięte.

Analiza danych

Czy na podstawie zaobserwowanych danych jesteś w stanie stwierdzić, czy odległość przeszkody od czujnika jest zgodna z oczekiwanym zasięgiem roboczym? Spróbuj zauważyć jak blisko przeszkody musi być czujnik, aby zmienił stan pinu GPIO na *high*. Czy zachowanie kodu jest zgodne z oczekiwaniami? Czy elementy na ekranie zmieniają się w odpowiedzi na zakrycie/odkrycie określonych czujników? Czy alarm włącza się po otwarciu przynajmniej jednego okna i czy kot mówi "Alarm!" w tej sytuacji?

Prezentacja wyników

Na tym etapie dzielimy się wynikami naszej pracy z innymi grupami. Czy wszystko działa poprawnie? Czy były jakieś trudności w ustawieniu całego toru? Czy były wprowadzane jakieś zmiany w kodzie? Jeśli tak, to jakie? Czy odczyty dotyczące tego, jak blisko musi znajdować się przeszkoda, aby port GPIO czujnika zmienił stan *high*, były spójne dla wszystkich czujników? Czy tak samo było w innych grupach?

1.2.4 Wnioski

Udało nam się stworzyć bardzo prosty obwód, który może służyć jako demonstracja instalacji bezpieczeństwa w budynku. Na tym etapie możemy wymienić się pomysłami z innymi grupami, co zostało zrobione w jaki sposób oraz w jakiej kolejności oraz wyjaśnić ewentualne wątpliwości.

1.2.5 Dodatkowe ćwiczenie (nieobowiązkowe)

W ramach ćwiczenia uzupełniającego można użyć multimetru, mierząc napięcie z czujników w trybie pod napięciem podczas zbliżania się do przeszkody. W ten sposób będziemy w stanie określić, przy jakim napięciu Raspberry Pi uznaje wejście za będące w stanie *high*. Możemy również spróbować zmienić pętlę główną użytą w kodzie z *repeat* na *forever*.



Erasmus+

2019-1-FR01-KA201-062281



STEMKIT
4SCHOOLS

1.3 Bibliografia i dodatkowe zasoby

Zasoby związane z tym planem lekcji: <https://www.vishay.com/docs/83760/tcrt5000.pdf>