



Wyższa Szkoła Handlowa w Radomiu

Systemy Wbudowane

Laboratorium 2

1 Cel zajęć

Celem ćwiczenia jest zapoznanie się z modulacją szerokości impulsu (Pulse Width Modulation – PWM). Efekt ten zostanie wykorzystany wraz z zewnętrznymi źródłami sygnału analogowego – potencjometrem i fotodiodą.

2 Teoria

Układy wbudowane są obecnie nieodzowną częścią niemal każdej większej konstrukcji układów elektryczno-elektronicznych. Nie sposób nie znaleźć ich dzisiaj w samochodach, sprzętach AGD, telefonach (które de facto są układami wbudowanymi złożonymi), parkomatami czy nawet w domofonach.

W powyżej wymienionych zastosowaniach układy spełniają różnorakie funkcje – monitorują temperaturę wewnątrz i na zewnątrz, sterują działaniem układów chłodzących bądź podających, przetwarzają sygnały analogowe na cyfrowe itd. Należy przy tym zauważyć, że układy te, chociaż działające w oparciu o logikę cyfrową (0/1), muszą również interpretować/wytwarzać sygnały analogowe. Powodów takiego działania układów jest wiele, jednak jeden zawsze najważniejszy i bezsprzeczny – my, jako ludzie, jesteśmy istotami analogowymi i nienaturalnym jest dla nas komunikowanie się za pomocą wartości binarnych.

Poniższe zadania dotyczyć będą właśnie interakcji pomiędzy wejściowymi sygnałami analogowymi, ich interpretacji przez układ cyfrowy oraz, na sam koniec, wyjścia analogowego, którego stan będzie wynikiem wspomnianych interpretacji.

3 Układ badawczy

W pierwszej kolejności należy utworzyć układ badawczy. W tym celu musimy posiadać:

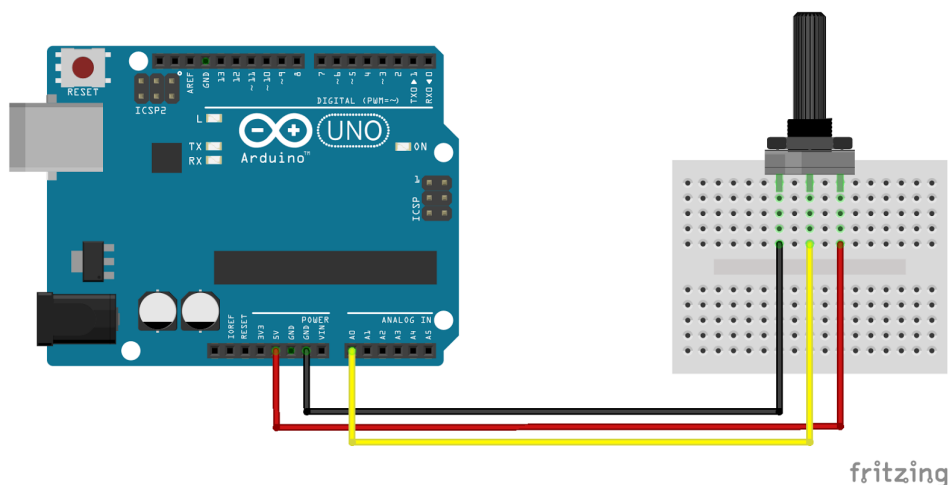
- układ Arduino Leonardo
- płytkę stykową pozwalającą na podłączenie elementów zewnętrznych
- potencjometr
- fotodioda
- dwie świecące diody (LED)
- jedna dioda RGB

3.1 Przykład podłączenia potencjometru do układu

Dołączony do zestawu potencjometr posiada 3 wyprowadzenia – masę, zasilanie (+5V) oraz wyjście analogowe. Podłączenie potencjometru może odbyć się bezpośrednio do odpowiednich portów na płytce i/lub poprzez płytkę stykową. Potencjometr, będący rezystorem tzw. suwakowym nie wymaga dodatkowego rezystora! Poniżej znajduje się obrazek przedstawiający możliwe podłączenie potencjometru do układu Arduino (na obrazku przedstawiony jest układ Uno jednak w ten sam sposób dokonuje się podłączenia pod układ Leonardo).

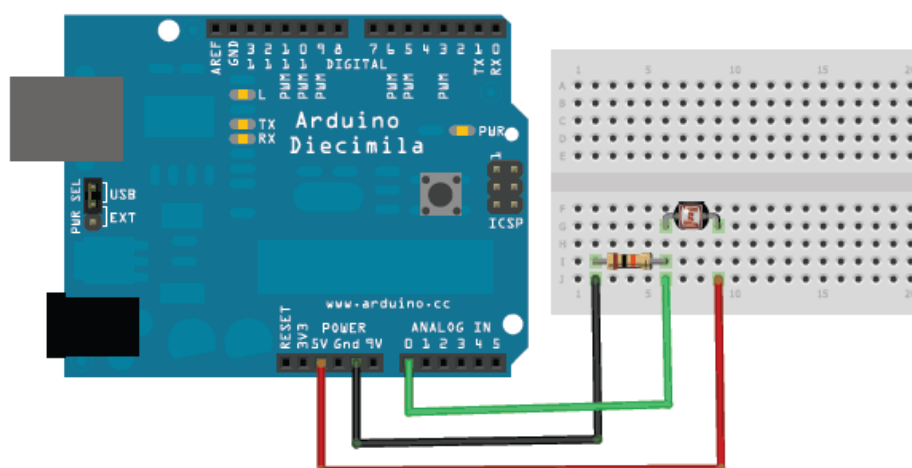
3.2 Przykład podłączenia fotodiody

Fotodioda, w przeciwieństwie do potencjometru, posiada jedynie dwa wyprowadzenia – zasilanie (+5V) oraz wyjście analogowe. Należy jednak pamiętać, że jak każdy element musi ona zostać podłączona do masy. Podłączenia



Rysunek 1: Schemat podłączenia potencjometru

tego dokonuje się poprzez instalację dodatkowego rezystora na złączu wyjścia. Dobór oporności zależy będzie od natężenia światła w miejscu, w którym będziemy dokonywać pomiarów. W przypadku nasłonecznionego miejsca możemy wykorzystać oporniki 100-300 ohm. W celu sprawdzenia tabeli oporności fotodiody dla pozostałego naświetlenia proszę udać się na stronę Using a photocell (dostępna w materiałach).

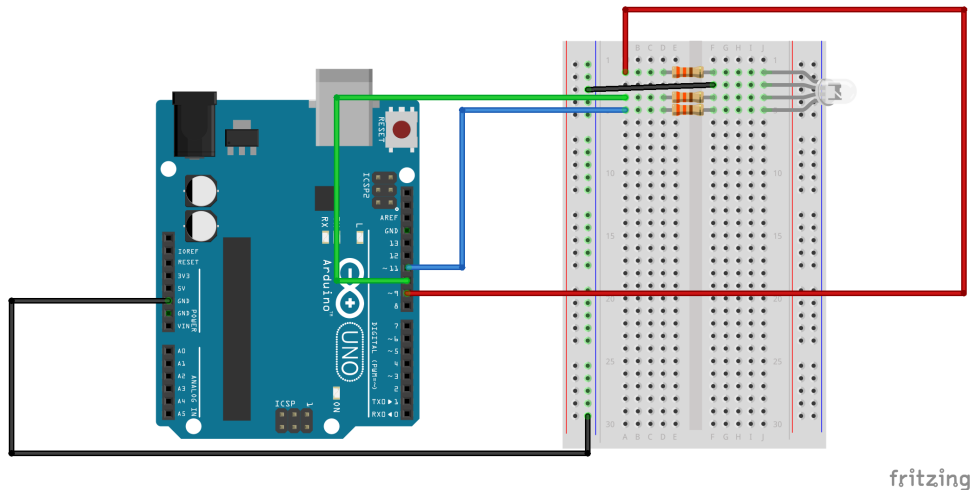


Rysunek 2: Schemat podłączenia fotodiody

3.3 Podłączenie diody świecącej RGB

Zasadniczo dioda ta niewiele różni się od standardowej diody świecącej. Różnica polega jedynie na ilości nóżek – dioda RGB posiada ich 4 (odpowiednio dla barwy czerwonej, zielonej, niebieskiej oraz jedną anodę). Tak jak w przypadku zwykłej diody musimy użyć oporników – po jednym dla każdego koloru. Oporność rezystorów może wynosić 220 ohm. Niektóre źródła podają by każdy kolor diody posiadał inną oporność w celu uzyskania lepszego efektu (dane składowe będą bardziej wygaszone, a tym samym lepiej będą oddziaływać w grupie).

Prozę zauważyć, że w tym wypadku anoda podłączona jest do masy. Jako zadanie dodatkowe proszę sprawdzić



Rysunek 3: Schemat podłączenia diody RGB

czy możliwe jest podłączenie diody do zasilania (+5V) i czy będzie ona wtedy działać poprawnie.

UWAGA Utworzenie układu to nie wszystko. Często może się bowiem zdarzyć, że chociaż wszystko wykonamy zgodnie ze sztuką, nasz układ nie będzie działał/będzie wyglądał na wadliwie wykonany. Przyczyn takiego stanu rzeczy może być kilka - wykorzystane złe piny, uszkodzona płytka stykowa, uszkodzone elementy czy też uszkodzone/nie dociśnięte przewody. Wreszcie powodem wadliwie działającego układu może być drobny błąd w kodzie. W celu przetestowania działania swojego układu oraz sprawdzenia jego reakcji na poszczególne stany można wyświetlić wartości, na których on operuje. Arduino nie posiada wyświetlacza w domyślnej konfiguracji dlatego jako kanał komunikacyjny wykorzystana zostanie konsola szeregowa. Jej działanie jest niezwykle proste – wystarczy zainicjować obiekt Serial i gotowe.

INFORMACJA W celu niniejszego ćwiczenia wykorzystywać będziemy jedynie wyświetlanie wiadomości. Obiekt Serial potrafi jednak znacznie więcej – szczegóły w odnośniku na końcu dokumentu.

4 Przykładowy kod

Zakładamy, że dioda RGB podłączona będzie do cyfrowych wyjść 3,5,6. Zakładamy też, że w przykładowym kodzie wykorzystamy potencjometr podłączony zostanie do analogowego złącza A0. Celem przykładowego kodu będzie uzyskiwanie barw na poszczególnych anodach wedle wartości pobranej z potencjometru.

```

1  #include <stdint>
2
3  enum
4  {
5      RED = 3,
6      GREEN = 5,
7      BLUE = 6
8  };
9  //wartosci potencjometru (0-255)
10 uint16_t pm = 0;
11
12 void pwmEmulation(uint16_t v);
13
14 void setup ()
15 {

```

```

16  pinMode (RED, OUTPUT);
17  pinMode (GREEN, OUTPUT);
18  pinMode (BLUE, OUTPUT);
19  //zapisujemy wartosci poczatkowe kolorow
20  digitalWrite (RED, LOW);
21  digitalWrite (GREEN, LOW);
22  digitalWrite (BLUE, LOW);
23  pm = analogRead (A0);
24  Serial.begin (9600);
25 }
26
27 void loop ()
28 {
29     pm = analogRead (A0);
30     Serial.print ("wartosc odczytana z potencjometru -> ");
31     Serial.println (pm);
32     //jedna z dwoch metod zapisu -> pelne PWM
33     analogWrite (RED, pm);
34     analogWrite (GREEN, pm + 15);
35     analogWrite (BLUE, pm + 50);
36     //druga opcja PWM (emulacja) w funkcji ponizej
37     delay (100);
38 }
39 //prosze zauwazyc, ze ten kod jest znacznie uproszczony (diody beda swiecic
    odcieniami szarosci)
40 //ponadto petla programu zajmuje dodatkowy czas (wartosc delay)
41 //rozbudowanie emulacji by byla rownowazna przedstawionemu w petli glownej
    dzialaniu zapisu
42 //analogowego wymaga zastosowania przerwan i rowniez bedzie opozniac dzialanie
    petli glownej programu
43 void pwmEmulation (uint16_t v)
44 {
45     digitalWrite (RED, HIGH);
46     digitalWrite (GREEN, HIGH);
47     digitalWrite (BLUE, HIGH);
48
49     delay (v);
50
51     digitalWrite (RED, LOW);
52     digitalWrite (GREEN, LOW);
53     digitalWrite (BLUE, LOW);
54 }

```

5 Zadania do wykonania

1. Utworzyć układ pozwalający na odczyt wartości z fotodiody oraz ustawienia potencjometru.
2. Napisać kod umożliwiający niezależny odczyt danych z fotodiody oraz potencjometru – dane można podejrzeć np. poprzez konsolę szeregową.
3. Podłączyć dwie diody świetlne i sprawdzić ich działanie prostym kodem (czy świecą); sprawdzić przyciemnianie kodu kopiując podany przykład.
4. Stworzyć kod umożliwiający sterowanie szerokością impulsu jednej diody przy pomocy fotodiody (gdzie przysłanianie fotodiody powoduje stopniowe zaciemnianie diody) oraz drugiej przy pomocy potencjometru
5. Podłączyć diodę RGB do układu i sprawdzić jej działanie

6. Napisać prosty kod pozwalający na sprawdzenie poprawności działania wspomnianej diody (wyświetlić kilka kombinacji RGB).
7. Stworzyć program, który pozwoli na sterowanie jasnością diody (fotodiody) oraz kolorami (potencjometr). Ponieważ operujemy jednym potencjometrem poszczególne kolory mają zmieniać swoje kolory w następujący sposób:
 - $R = \text{wartosc_potencjometr} + (\text{wartosc_fotodioda} * 20\%)$
 - $G = \text{wartosc_potencjometr}$
 - $B = \text{wartosc_fotodioda} + (\text{wartosc_potencjometr} * 35\%)$

DLA STUDENTÓW STUDIÓW NIESTACJONARNYCH:

8. Należy sporządzić sprawozdanie z przebiegu ćwiczenia według podanego szablonu (dostępny na stronie).

6 Materiały wykorzystane przy opracowaniu

<http://www.microchip.com/wwwproducts/en/ATmega32u4>

<http://www.instructables.com/id/Step-by-Step-Guide-to-the-Arduino-Leonardo/>

<https://www.arduino.cc/en/Hacking/BuildProcess>

<https://processing.org>

<http://blog.nettigo.pl/post/85619491578/czemu-powsta\T1\lo-arduino>

<http://wiring.org.co>

http://www.uni-koeln.de/phil-fak/muwi/ag/praktikum/assembler_arduino.pdf

http://blog.codebender.cc/wp-content/uploads/2014/03/blink_.png

http://breakoutjs.com/examples/getting_started/images/hello_world_schematic.png