

# “Efekty świetlne”

## Sprawozdanie wykonania projektu

Piotr Kucharski, Dominik Zabłotny

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

Informatyka, II rok - 2017/2018

Grupa: środa, 11:00

### 1. Schemat połączeń płytki i dodatkowych modułów

W celu poprawnego zestawienia połączenia elementów z płytką Arduino Uno należy podłączyć linijkę cyfrowych diod RGB oraz enkoder z przyciskiem za pomocą kabli według schematu przedstawionego w tabeli:

Urządzenie	Piny na urządzeniu	Piny Arduino Uno
Linijka diod RGB	GND	GND
	DIN	A0
	4-7VDC	5V
	GND	GND
Enkoder z przyciskiem	CLK	5
	DT	6
	SW	7
	+	3.3V
	GND	GND

### 2. Opis algorytmu

Projekt opiera się na głównej, nieskończonej pętli, której akcje są sterowane poprzez sprawdzanie oraz ustawianie odpowiednich flag wykonania programu. W zależności od tych zmiennych ustawiany jest odpowiedni program świetlny oraz włączany jest tryb edytowania ilości poziomów jasności.

Program rozpoczyna pracę od standardowych wartości zmiennych sterujących zadeklarowanych w stałych preprocesorowych o nazwie “EFFECT”, “INIT\_K” oraz

“INITIAL\_U3\_DELAY”. Domyślnie, program uruchamia się na zerowym efekcie świetlnym z ilością poziomów świetlnych ustawionym na  $2^2$  oraz ze standardową szybkością zmiany świecącej diody w programie “U3”.

Wykonany projekt oferuje wyświetlanie dziewięciu programów świetlnych w następującej kolejności:

- *R* - wszystkie diody ustawione na kolor czerwony,
- *G* - wszystkie diody ustawione na kolor zielony,
- *B* - wszystkie diody ustawione na kolor niebieski,
- *RG* - wszystkie diody ustawione na kolor żółty,
- *RB* - wszystkie diody ustawione na kolor różowy,
- *GB* - wszystkie diody ustawione na kolor cyjankowy,
- *U1* - na przemian migające diody o kolorze niebieskim i czerwonym,
- *U2* - pasek świecący wszystkimi diodami przechodzący przez całą gamę kolorów,
- *U3* - niebieski punkt przechodzący przez cały pasek, „odbijający” się od końców paska ledowego zmieniający świecąca diodę o czas ustawiany poprzez obrót enkodera z możliwością zatrzymania punktu na wybranej diodzie poprzez kliknięcie enkodera.

Przejście pomiędzy programami odbywa się poprzez krótkie naciśnięcie przycisku enkodera. Dłuższe jego przytrzymanie spowoduje wejście w tryb edycji parametru *K* stanowiącego ilość poziomów świetlnych możliwych do wyświetlenia dla danej diody. Ilość poziomów oblicza się wzorem  $n = 2^K$  gdzie  $K \in \{2, 3, 4, 5, 6\}$ . W trybie edycji tego współczynnika kolejne wartości są wyświetlane poprzez zapalenie się odpowiedniej ilości białych diod, które reguluje się poprzez przekręcanie enkodera. Krótkie naciśnięcie przycisku w trybie edycji *K* spowoduje odrzucenie zmian, zaś dłuższe przytrzymanie zatwierdzi zmiany.

Wyżej opisany sposób działania obsługi enkodera działa dla wszystkich programów, oprócz *U3*. Wówczas, krótkie wciśnięcie przycisku enkodera zatrzyma poruszanie się niebieskiego punktu po linii diod a dłuższe naciśnięcie spowoduje przejście do następnego programu (w tym przypadku do pierwszego programu: *R*).

### 3. Opis działania programów użytkownika

Zgodnie z wytycznymi płytki oferuje wyświetlanie trzech autorskich programów:

- *U1* - program wyświetlający na przemian niebieskie i czerwone diody, w czasie rzeczywistym sprawdzający czy za pomocą enkodera została zmieniona jasność.
- *U2* - program wyświetlający efekt ‘tęczy’, czyli stopniową zmianę kolorów. Korzystając z pętli ‘for’ uzyskujemy płynny efekt przejścia po kolorach. W trakcie trwania programu w czasie rzeczywistym sprawdzane jest czy za pomocą enkodera została zmieniona jasność.
- *U3* - program wyświetlający ‘odbijającą’ się od krawędzi paska ledowego niebieską diodę, której ruch wykorzystując enkoder może być zatrzymywany i wznowiany przez użytkownika. Korzystając z dwóch pętli ‘for’ uzyskujemy efekt płynnego przejścia niebieskiej diody od jednej krawędzi paska ledowego do drugiej. Program wykorzystuje zmienioną funkcję przycisku enkodera.

## 4. Ważne sekcje kodu programu oraz wykorzystane biblioteki

W programie zostały użyte dwie biblioteki: **Adafruit\_NeoPixel.h** dla obsługi paska ledów RGB, oraz **RotaryEncoder.h** w celu łatwego odczytywania przekręcenia pokrętła enkodera. Biblioteki można pobrać bezpośrednio z poziomu Arduino IDE za pomocą "Menedżera bibliotek".

Program opiera się na flagach kontrolujących przepływ wykonania programu. W celu zmniejszenia zużycia pamięci wszystkie 6 flag zostało zapisanych w jednej zmiennej jednobajtowej bez znaku: **unsigned char flags**. Kolejność flag wraz z opisem od najbardziej znaczącego bitu:

- 0 → *null* - nie używany bit
- 1 → *WAS\_CHANGE* - flaga odnotowuje czy użytkownik wykonał jakąkolwiek operację sterowania. Jest wykorzystywana do zapobiegania niepotrzebnego dalszego wykonywania pętli próbkującej odczyt wejścia.
- 2 → *IS\_UP* - flaga określa kierunek podróży niebieskiego punktu w programie U3
- 3 → *LIGHT* - określa czy niebieski punkt powinien się świecić czy nie
- 4 → *U3* - flaga oznacza, że program U3 jest odtwarzany.
- 5 → *U2* - flaga oznacza, że program U2 jest odtwarzany.
- 6 → *U1* - flaga oznacza, że program U1 jest odtwarzany.
- 7 → *IS\_CHANGING\_K* - flaga determinuje, czy program jest w trybie ustawiania parametru K.

Odczytywanie tych flag odbywa się za pomocą funkcji **bitRead**, ustawianie oraz zerowanie flag ustawiane jest za pomocą odpowiednio **bitSet** oraz **bitClear**.

Wspomniana wcześniej pętla próbkowania wejścia użytkownika działa na zasadzie odczytywania obrotu enkodera oraz przycisku w pętli z próbkowaniem określonym stałą preprocesora **SAMPLING\_RATE**. Zostało to wprowadzone w celu zwiększenia czasu procesora przeznaczonego na odczytywanie wejścia.

Bieżący program jest określony za pomocą zmiennej globalnej **unsigned char led\_program**, które zwiększa się za każdą zmianą programu oraz poddawana jest działaniu modulo 9, aby nigdy nie przekroczyła tej wartości. Później funkcja **set\_program** uruchamia odpowiednią procedurę bazując na tej zmiennej.

Referencja do paska ledowego jest umieszczona w globalnej zmiennej **Adafruit\_NeoPixel strip**, zaś enkoder jest odczytywany z poziomu zmiennej **RotaryEncoder encoder**.

Odróżnianie krótkiego przyciśnięcia przycisku od długiego polega na każdorazowym dodaniu 0.5 lub 0.1 do zmiennej **button\_down\_counter**. Różnica którą wartość dodać wynika z różnego obciążenia procesora podczas wykonywania poszczególnych programów - dla programu U3 z racji wykonywania dużej ilości pętli należy dodać większą wartość w celu mniejszego czasu wciskania przycisku aby zmienić program (długie przytrzymanie).

## 5. Uruchomienie projektu oraz pozostałe informacje

Kompilacja oraz uruchomienie programu nie odbiega od standardowej kompilacji. Należy do przygotowanej płytki Arduino podłączyć komputer za pomocą kabla USB oraz z poziomu Arduino IDE, po uprzednim skonfigurowaniu portu i rodzaju płytki, skompilować i wgrać program do urządzenia. Projekt jest łatwy do rozszerzenia o kolejne programy świetlne poprzez napisanie odpowiedniej funkcji sterującej paskiem ledowym oraz rozszerzenie ciała polecenia *switch* w funkcji ***void set\_program()*** oraz powiększeniem stałej preprocesora ***N\_PROGRAMS***.

Kolejne wersje projektu powstawały w oparciu o naukę programowania w środowisku Arduino. Po uprzednim podzieleniu obowiązków w zespole, każdy członek skupił się na swoim zadaniu tworząc swoje małe projekty, sprawdzając ich działanie. Następnie małe projekty zostały scalone w jeden duży projekt finalny, dopasowany do specyfikacji.

Do sprawozdania zostaje dołączony plik "*src\_KucharskiZablotny.ino*" zawierający kod źródłowy projektu.