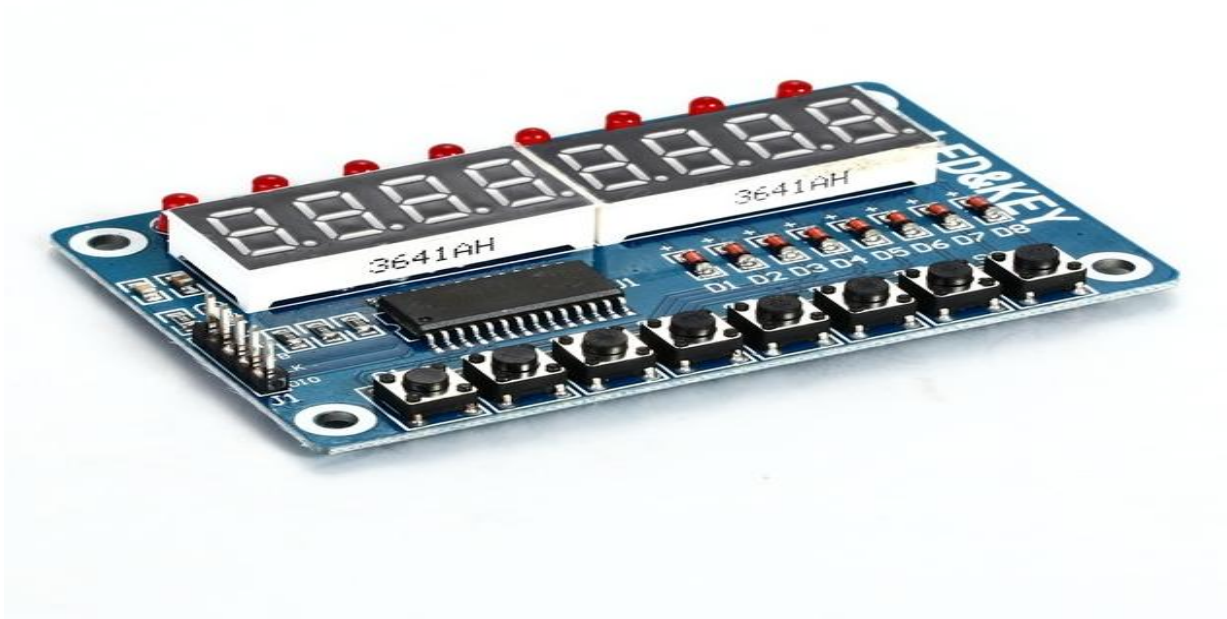


## Mikrokontrolery – wytyczne do projektów

### Temat 1 – „Interfejs do bomby”



Program ma imitować proces ustawiania opóźnienia „wybuchu bomby” i zmianę szybkości odliczania w dół. Po uruchomieniu program ma wyświetlić **COUNT** zer począwszy od skrajnego prawego segmentu. Dłuższe naciśnięcie przycisku enkodera powoduje wejście do trybu edycji licznika – obrót pokrętła powinien umożliwiać wybór odpowiedniej cyfry

(segmentu), zaś krótkie naciśnięcie przycisku skutkuje zmianą wyświetlanej wartości w trakcie obracania pokrętle. Aktualnie wybrany segment sygnalizowany jest migotaniem w trakcie edycji. Krótkie naciśnięcie przycisku zmienia tryb edycji (wybór segmentu/wybór wartości). Dłuższe naciśnięcie powoduje wyjście z edycji – na wszystkich segmentach wyświetlane są odpowiednie wartości bez migotania. Krótkie naciśnięcie uruchamia odliczanie w dół – kolejne naciśnięcie zatrzymuje proces odliczania. W trakcie odliczania obrót pokrętle powinien skutkować zmianą szybkości odliczania (lewo=wolniej, prawo=szybciej) – prędkość minimalna to zmiana wartości o 1 w ciągu sekundy, prędkość maksymalna powinna zapewnić dojście do 0 od wartości maksymalnej (99999999 dla COUNT=8, 9999 dla COUNT=4, itp.) w ciągu 60 sekund. Osiągnięcie wartości 0 w trakcie odliczania powinno być zasygnalizowane migotaniem wszystkich segmentów – krótkie naciśnięcie wyłącza migotanie a program przechodzi do stanu początkowego. W trakcie odliczania wejście do trybu edycji jest niemożliwe.

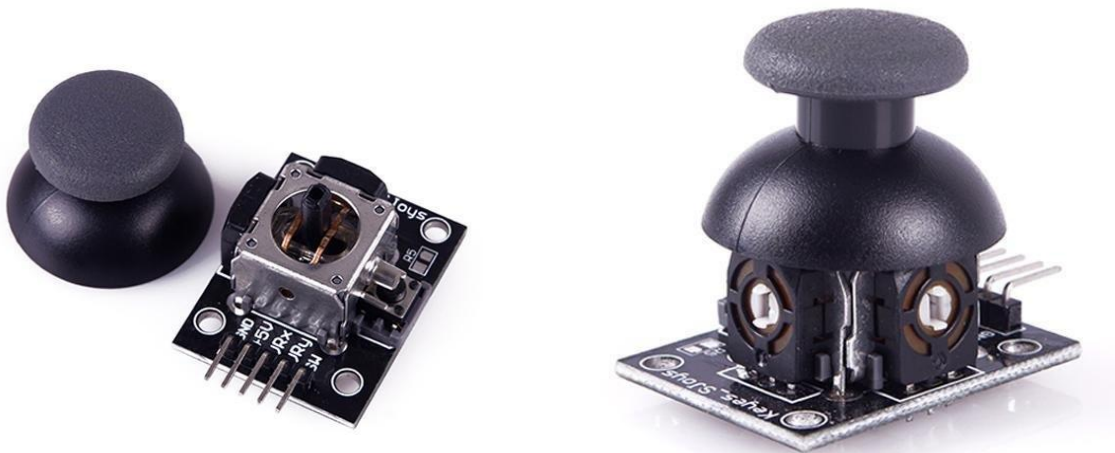
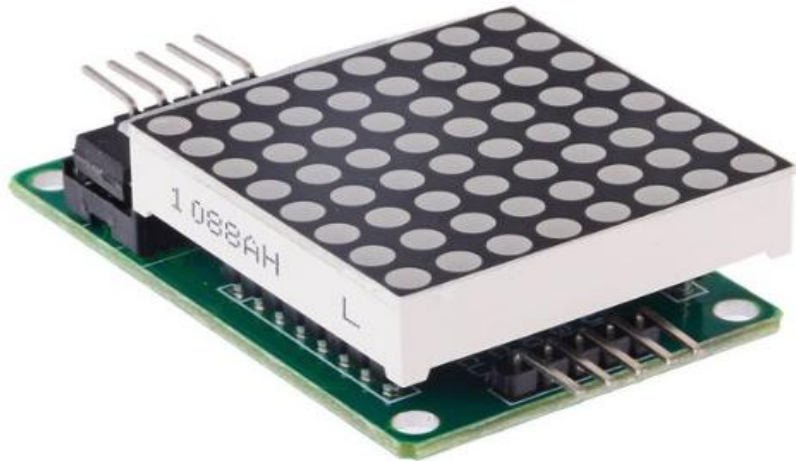
Po starcie program powinien wykorzystać wartości domyślne zapisane w sposób umożliwiający ustalenie ich wartości poprzez łatwą modyfikację kodu źródłowego na etapie demonstracji działania programu (edycja przed wgraniem kodu do mikrokontrolera).

Parametry (ustalane przed kompilacją kodu):

- CODE** - ciąg maksymalnie ośmiu cyfr w postaci liczby lub łańcucha znaków (wartość początkowa="00000000")
- COUNT** - liczba wykorzystanych segmentów [1..8] (wartość początkowa=8)

**Punkt startowy:** Google „arduino led key tm1638”, „arduino encoder”

## Temat 2 – „Ping pong”



Program ma imitować grę w ping-ponga (tenis stołowy) ze ścianą. Ruch joysticka w pionie pozwala na przesuwanie rakiетки (też w pionie), ruch w poziomie (pojedyncze wychylenia) powinien zmieniać szybkość poruszania się piłeczki (lewo=zwolnienie, prawo=przyspieszenie). Rakiетка porusza się po prawej krawędzi wyświetlacza – ma wielkość **RSize** pikseli i w trakcie ruchu może się całkowicie znaleźć poza wyświetlaczem (stołem). Po uruchomieniu programu wyświetlana jest tylko rakiетка. Krótkie naciśnięcie joysticka rozpoczyna pierwszą grę – piłka pojawia się na lewej krawędzi w środku wysokości matrycy

(losowy wybór czwartego lub piątego piksela w kolumnie – prawdopodobieństwa po 50%). Od lewej krawędzi piłka porusza się w losowo wybranym kierunku (ukośnie w górę, poziomo, ukośnie w dół – prawdopodobieństwa po 33,3%), ewentualne zmiany toru piłki w wyniku odbicia od górnej lub dolnej krawędzi mają być dokonywane zgodnie z zasadami fizyki (kąt padania jest równy kątowi odbicia). Aby uznać piłkę za odbitą od rakiety, w momencie osiągnięcia przez piłkę prawej krawędzi musi ona pokrywać się z jednym z pikseli tworzących raketę. Zmiana toru piłki po odbiciu się od rakiety jest losowa (40%-zgodnie z zasadami fizyki, 20%-ukośnie w górę, 20%-poziomo, 20%-ukośnie w dół). Brak odbicia oznacza utratę punktu. Po stracie **POINTS** punktów następuje zakończenie gry i wyświetlenie liczby odbić w czytelnej formie na migającym wyświetlaczu (np. zapis binarny, grupy diod wyświetlające poszczególne cyfry dziesiętne wyniku, skrolowana wartość dziesiętna). Krótkie naciśnięcie joysticka powoduje przejście do stanu początkowego. Dłuższe przyciśnięcie w trakcie gry (poza wyświetlaniem wyniku) powinno skutkować cykliczną zmianą rozmiaru rakiety.

Minimalna i maksymalna prędkość piłki powinny zostać tak dobrane w stosunku do prędkości poruszania się rakiety aby zapewnić jak największą „grywalność” – prędkość poruszania się rakiety ma być stała, zmianie ma podlegać jedynie prędkość piłki!

Po starcie program powinien wykorzystać wartości domyślne zapisane w sposób umożliwiający ustalenie ich wartości poprzez łatwą modyfikację kodu źródłowego na etapie demonstracji działania programu (edycja przed wgraniem kodu do mikrokontrolera).

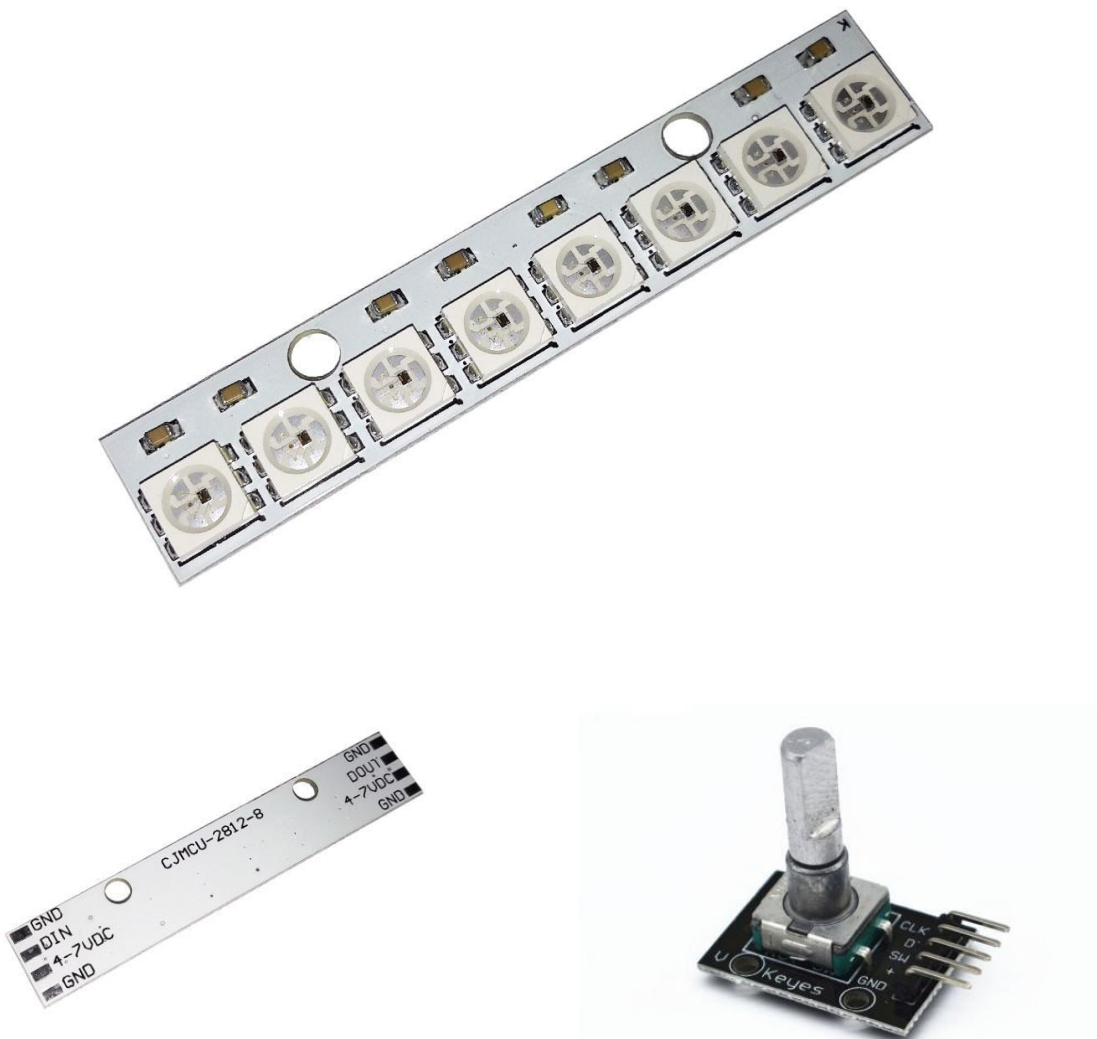
Parametry (ustalane przed kompilacją kodu):

**RSIZE**            - wielkość rakiety [1..5] (wartość początkowa=3)

**POINTS**          - liczba nieudanych prób odbić kończąca grę (wartość początkowa=5)

**Punkt startowy: Google „max7219 arduino”, „arduino joystick”**

### Temat 3 – „Efekty świetlne”



Program ma wyświetlać różnorodne kolorowe efekty świetlne wykorzystując linijkę cyfrowych diod RGB, zaś elementem sterującym jest enkoder z przyciskiem. Jasności świecenia poszczególnych diod powinny się różnić zgodnie z poniższymi zasadami:

- $L_{\min}$  – poziom minimalny (minimalne wartości składowych RGB pozwalające zauważyć świecenie diody - ustalenia eksperymentalnie)
- $L_{\max}$  – poziom maksymalny (składowe RGB mają wartość 255)

- **N** – liczba poziomów jasności zmieniających się skokowo od  $L_{\min}$  do  $L_{\max}$  ( $N=2^K$ ,  $K=2..6$ ) wykorzystanych do utworzenia wirtualnego ciągu diod o długości  $2*N$ , których jasności zmieniają się w następujący sposób:

$$I_0, I_1, \dots, I_{N-1}, I_{N-1}, I_{N-2}, \dots, I_1, I_0 \quad I_0=L_{\min}, I_{N-1}=L_{\max}$$

Po rozpoczęciu działania programu diody mają jasności odpowiadające pierwszym 8 elementom powyższego ciągu, zaś obracanie pokrętkiem enkodera powinno powodować zmianę jasności diod (wyświetlanie kolejnych diod wirtualnych – zawsze wyświetlanych jest osiem sąsiednich z  $2*N$  możliwych). Krótkie naciśnięcie przycisku enkodera powinno zmienić rodzaj wyświetlanego efektu (koloru) w następującej kolejności: R (diody świecą na czerwono), G, B, RG, RB, GB, U1, U2, U3 (łączna liczba efektów ma być równa 9). Efekty U1, U2, U3 są efektami zdefiniowanym przez autorów – jasności i kolory diod mogą być dowolne, ale działanie przynajmniej jednego z efektów użytkownika powinno uwzględniać obracanie pokrętkiem enkodera. Dłuższe naciśnięcie przycisku enkodera powinno umożliwiać zmianę wartości parametru **K** z przedziału  $[2..6]$  – sama wartość powinna być sygnalizowana świeceniem odpowiedniej liczby diod w kolorze białym. Ponowne dłuższe naciśnięcie przycisku winno skutkować uaktualnieniem wartości parametru **K** i powrotem do ostatnio wyświetlanego efektu (stanu sprzed edycji), krótkie naciśnięcie oznacza wyjście z edycji bez zmiany parametru **K**.

Uwzględnienie w efektach użytkownika wartości **K** nie jest konieczne, ale będzie dodatkowym atutem.

Po starcie program powinien wykorzystać wartości domyślne zapisane w sposób umożliwiający ustalenie ich wartości poprzez łatwą modyfikację kodu źródłowego na etapie demonstracji działania programu (edycja przed wgraniem kodu do mikrokontrolera).

Parametry (ustalane przed kompilacją kodu):

**K** - wartość początkowa = 2

**EFFECT** - numer efektu = 1 (1=R, 2=G, 3=B, 4=RG, 5=RB, 6=GB, 7=U1, 8=U2, 9=U3)

**Punkt startowy:** Google „ws2812 arduino”, „arduino encoder”