

SPRAWOZDANIE						PROSZĘ PODAĆ NR GRUPY:						
						ZIASS1	3	5	1	2	IO	
IMIĘ	NAZWISKO	Temat ćwiczenia zgodny z wykazem tematów:				PONIŻEJ PROSZĘ PODAĆ TERMIN ZAJĘĆ:			ROK:			
MICHAŁ	WARSZAWSKI	Matryce diodowe Sterowane bezpośrednio z portów procesora i za pomocą interfejsu SPI				PN	WT	SR	CZ	PT	SB	ND
						2023 r.						
						GODZINA ROZPOCZĘCIA ZAJĘĆ:						
						11 : 30						

Wprowadzenie teoretyczne:

### Opisz dostępne rodzaje matryc. Opisz interfejs SPI

W Arduino, matryce LED są powszechnie używane do wyświetlania grafiki, tekstu lub innych informacji w postaci diod elektroluminescencyjnych (LED). Matryce LED mogą mieć różne rozmiary i rodzaje, a ich kontrola zwykle wymaga specjalnych bibliotek i technik multiplexingu.

#### ➤ Matryce LED jednokolorowe:

- Matryce jednokolorowe zawierają diody LED o jednym kolorze, na przykład czerwonym, zielonym lub niebieskim. Każda dioda może być włączona lub wyłączona, co umożliwia wyświetlanie prostych wzorów lub napisów w jednym kolorze.

#### ➤ Matryce LED wielokolorowe (RGB):

- Matryce wielokolorowe posiadają diody RGB (czerwona, zielona, niebieska) w każdym punkcie pikselowym. To pozwala na wyświetlanie pełnej gamy kolorów poprzez kontrolę intensywności poszczególnych składowych koloru. Sterowanie takimi matrycami wymaga większej liczby pinów Arduino i bardziej zaawansowanych technik programowania.

#### ➤ Matryce LED dwukolorowe (Bicolor):

- Matryce dwukolorowe posiadają diody dwukolorowe (na przykład czerwono-zielone) w każdym punkcie pikselowym. Mogą one emitować jeden z dwóch kolorów, co umożliwia wyświetlanie prostych animacji lub informacji w dwóch kolorach.

#### ➤ Matryce LED siedmiosegmentowe:

- Matryce siedmiosegmentowe to specyficzny rodzaj matryc LED, w którym każda cyfra jest reprezentowana przez kombinację siedmiu diod LED, układanych w taki sposób, aby tworzyć cyfry od 0 do 9 oraz niektóre litery. Są one często używane do wyświetlania liczb i prostych znaków.

#### ➤ Matryce LED dot-matrix:

- Matryce LED dot-matrix to matryce o większej ilości diod LED, które można programować do wyświetlania różnych wzorów i znaków. Kontrola takiej matrycy wymaga zastosowania techniki multiplexingu, co pozwala na wykorzystanie mniejszej liczby pinów mikrokontrolera.

Kontrola matryc LED w Arduino zazwyczaj odbywa się poprzez biblioteki, takie jak "LedControl" czy "MaxMatrix", które ułatwiają obsługę diod w matrycach.

**Interfejs SPI** (Serial Peripheral Interface) w Arduino to protokół komunikacyjny umożliwiający komunikację pomiędzy mikrokontrolerem a różnymi urządzeniami peryferyjnymi, takimi jak czujniki, ekrany LCD, karty pamięci czy inne mikrokontrolery. Jest to szybki interfejs, który umożliwia jednoczesną komunikację z wieloma urządzeniami. Poniżej znajduje się szczegółowe omówienie interfejsu SPI w kontekście platformy Arduino.

#### Zasada Działania:

Interfejs SPI opiera się na komunikacji synchronicznej, co oznacza, że dane są przesyłane w określonym rytmie zegarowym. W Arduino, linie SPI obejmują:

- **MISO (Master In Slave Out):**
  - Jest to linia, po której dane są przesyłane z urządzenia podrzędnego (Slave) do urządzenia nadrzędnego (Master).
- **MOSI (Master Out Slave In):**
  - Linia, po której dane są przesyłane z urządzenia nadrzędnego do urządzenia podrzędnego.
- **SCK (Serial Clock):**
  - Jest to linia zegara, która synchronizuje przesyłanie danych między Masterem a Slave'em.
- **SS/CS (Slave Select/Chip Select):**
  - Ta linia jest używana do wybierania konkretnego urządzenia Slave, z którym Master chce komunikować się w danym momencie.

**Funkcje Biblioteki SPI:**

<code>SPI.begin();</code>	- Inicjalizuje interfejs SPI
<code>SPI.setClockDivider(divider);</code>	- Ustawia dzielnik zegara. Możliwe wartości to np. <code>SPI_CLOCK_DIV2</code> ,
<code>SPI.setDataMode(mode);</code>	- Ustawia tryb danych, na przykład <code>SPI_MODE0</code> , <code>SPI_MODE1</code> itd.
<code>SPI.setBitOrder(order);</code>	- Ustawia kolejność bitów, na przykład <code>MSBFIRST</code> lub <code>LSBFIRST</code>
<code>SPI.transfer(data);</code>	- Przesyła jedno bajt danych w obie strony

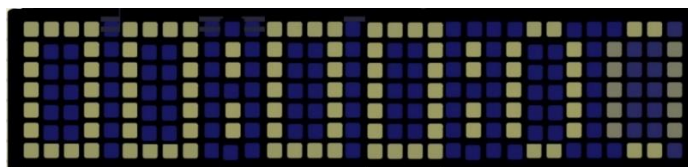
Interfejs SPI w Arduino zapewnia elastyczność i wydajność w komunikacji z różnymi peryferiami, co czyni go popularnym wyborem w projektach związanych z mikrokontrolerami.

## Zadanie 1

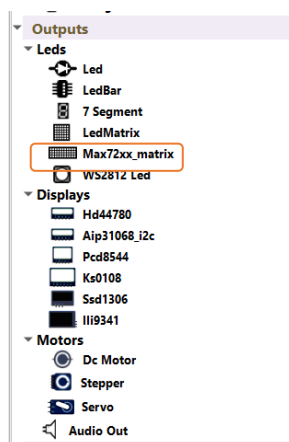
Podłącz do płyty Arduino Mega matrycę diodową LedMatrix zmieniając jej rozmiar z domyślnego 8x8 na 8X33 a następnie podobnie jak w zadaniu 5 zaprogramuj zegar odliczający sekundy miny i godziny. Proces odliczania przyspiesz dziesięciokrotnie. Zadanie należy wykonać z użyciem jednej matrycy LedMatrix a liczby na zegarze mają pojawiać się w kolejności odliczania. Oznacza to, że jeżeli nie są odliczane minuty i godziny zegar nie wyświetla tych liczb.

Zwróć uwagę, że im więcej cyfr wyświetlisz na matrycy tym będą one słabiej świecić. Spróbuj rozwiązać ten problem. Format (kształt cyfry pozostawiam do wyboru).

**UWAGA! Tym razem nie można wykorzystywać gotowych bibliotek.**



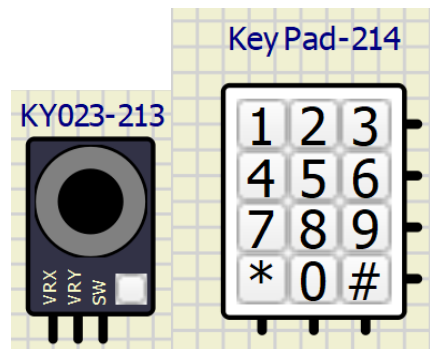
Rys. 1 Przykład zegara.



Rys. 2 Menu „Outputs” z zaznaczonym elementem LedMatrix

## Zadanie 2

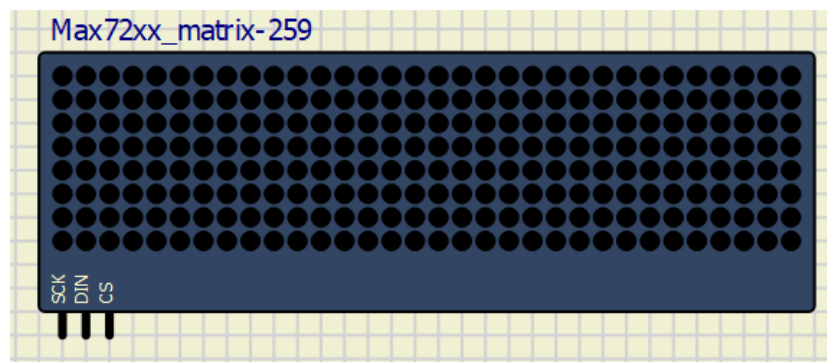
Po wykonaniu zadania pierwszego zmień ustawienia matrycy LedMatrix i teraz zamiast ustawienia domyślnego 8x8 zmień jej rozmiary na 33x8 a następnie zaprojektuj grę MINI TETRIS. Do sterowania grą wykorzystaj klawiaturę KeyPad lub Joystick KY-023. Manipulator w prawo przesunięcie elementu w prawo. Manipulator w lewo przesunięcie elementu w lewo. Manipulator w górę obrót elementu w prawo. Manipulator w dół obrót elementu w lewo. Przy wykorzystaniu klawiatury można samodzielnie wybrać klawisze.



Rys. 3 Joystick i Klawiatura

### Zadanie 3

Powtórz zegar z zadania 1 z użyciem matrycy Max 72xx\_matrix-259.



Rys. 4 Wyświetlacz matrycowy Max 72xx

**W tym zadaniu można używać bibliotek.**

Ponieważ matryca posiada 32 kolumny pierwsza cyfra z prawej będzie przyjmowała maksymalną wartość „1” czyli zegar będzie odliczał tylko 12 godzin.

Wykorzystaj klawiaturę Keypad przedstawioną na rysunku 3 tak aby możliwe było przestawianie cyfr w zegarze.

# wybór: godziny, minuty, sekundy. Po przyciśnięciu powinny migać cyfry.

\* zatwierdzenie wyboru

Z klawiatury numerycznej można pisywać cyfry.

### Wnioski:

W ramach zadania pierwszego tego zadania matryca diodowa LedMatrix została podłączona do płyty Arduino Mega, a jej rozmiar zmieniono na 8x33. Następnie zaimplementowano zegar, który odliczał czas w trybie przyspieszonym. Wyzwaniem było zoptymalizowanie wyświetlania cyfr zegara, aby uniknąć problemu słabego świecenia przy wyświetlaniu większej liczby cyfr. Kod programu zawiera definicje pinów, macierzy dla poszczególnych cyfr, oraz funkcje do wyświetlania czasu na matrycy LED. Dodatkowo, aby uzyskać przyspieszone odliczanie, wykorzystano zmienną interval, określającą prędkość w milisekundach.

Po zakończeniu pierwszego zadania, matryca LedMatrix została ponownie skonfigurowana, tym razem jako 33x8. Następnie należało zaprojektować grę Mini Tetris, wykorzystując do sterowania klawiaturę KeyPad. Klawisze kierunkowe wykorzystano do poruszania klockami, a dodatkowe klawisze do obrotu w lewo i w prawo.

W ostatnim zadaniu zegar został powtórzony, tym razem wykorzystując matrycę Max 72xx\_matrix-259. W przeciwieństwie do poprzedniego zegara, tym razem możliwe było korzystanie z gotowych bibliotek. Matryca posiadała 32 kolumny, co ograniczało zegar do odliczania maksymalnie 12 godzin. Wprowadzono także obsługę klawiatury Keypad, umożliwiającą wybór trybu (godziny, minuty, sekundy) oraz przestawianie cyfr w zegarze.

W trakcie ćwiczenia poruszono różne aspekty sterowania matrycami LED, począwszy od bezpośredniego sterowania z portów procesora, poprzez implementację zegara na matrycy LedMatrix, aż po stworzenie gry Mini Tetris. Ostatnie zadanie wprowadziło element interakcji z użytkownikiem poprzez klawiaturę, co poszerzyło zakres możliwości sterowania matrycą LED. Warto zauważyć, że różne rozmiary matryc oraz różne metody sterowania wymagają indywidualnego podejścia do implementacji i optymalizacji kodu.