

**Politechnika  
Warszawska**

**Technika mikroprocesorowa II  
PRACA ZALICZENIOWA**

**Kajetan Jeznach**

Prowadzący:

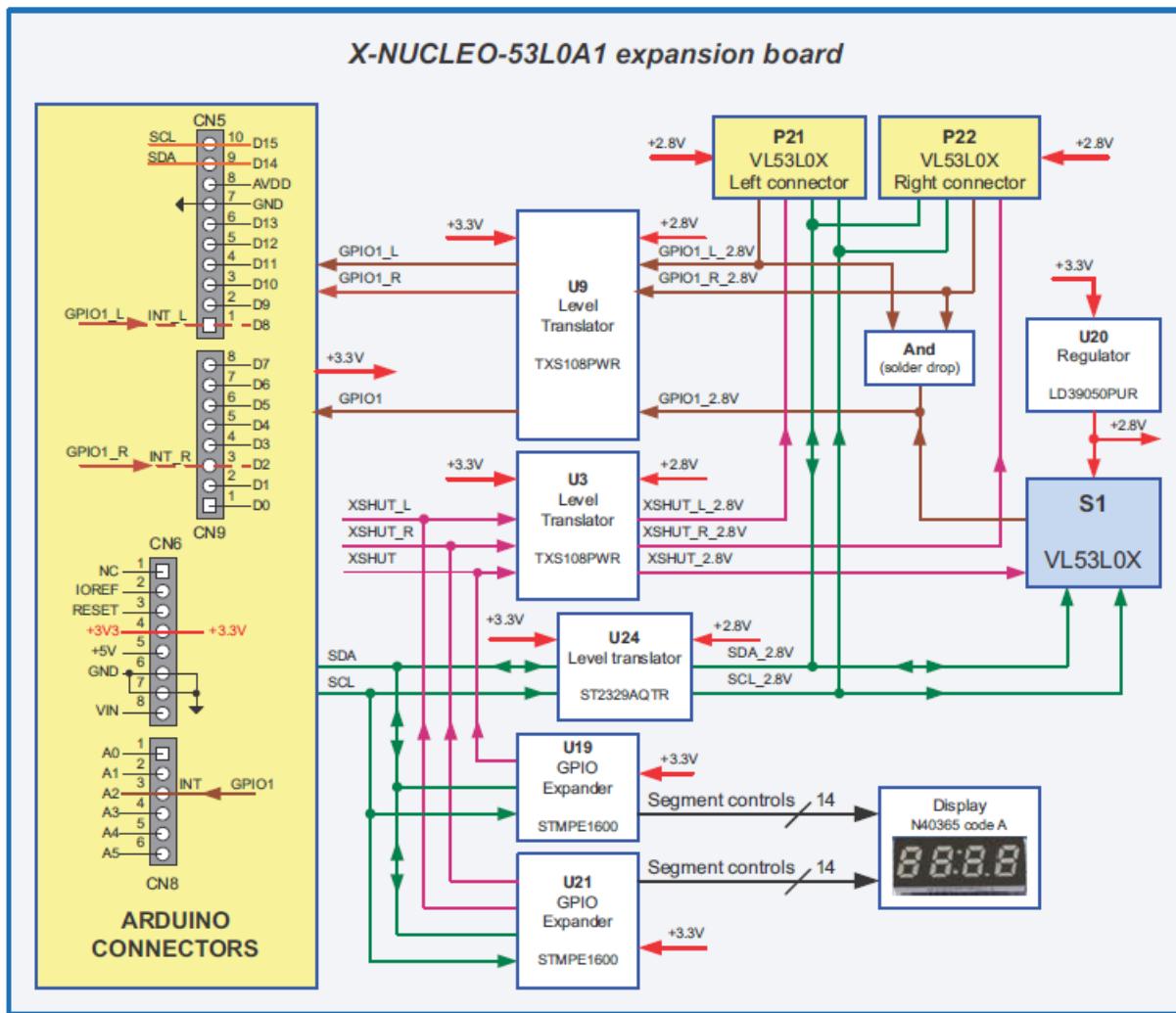
Data oddania: **21.01.2022**

## Spis treści

<b>1</b>	<b>Opis zadania</b>	<b>3</b>
<b>2</b>	<b>Część odpowiedzialna za próbkowanie temperatury</b>	<b>4</b>
<b>3</b>	<b>Część odpowiedzialna za komunikację I2C</b>	<b>4</b>
3.1	Procedura I2C . . . . .	4
3.2	Transmisja . . . . .	5
<b>4</b>	<b>Funkcja główna</b>	<b>5</b>
<b>5</b>	<b>Część odpowiedzialna za wyświetlanie liczb</b>	<b>5</b>
<b>6</b>	<b>Zastosowanie</b>	<b>6</b>
<b>7</b>	<b>Kalibracja wyświetlonej temperatury</b>	<b>7</b>

# 1 Opis zadania

Zadaniem zaliczeniowym było wykorzystanie magistrali I<sub>2</sub>C do obsługi wbudowanego do płytki X-NUCLEO-53L0A1 wyświetlacza 7-segmentowego. Wyświetlana wartość na wyświetlaczu to temperatura mikroprocesora oraz znak °C. Wspomnianą temperaturę odczytujemy za pomocą przeskalowania odczytanego napięcia z przetwornika ADC. W tym celu odpowiednio zmodyfikowano kod z poprzedniego semestru. Na rysunku 1. przedstawiono schemat blokowy połączeń płytka.



Rysunek 1: Schemat płytka rozszerzeń X-NUCLEO-53L0A1

## 2 Część odpowiedzialna za próbkowanie temperatury

Kod do obliczania temperatury wykorzystywany w poprzednim semestrze został zmodyfikowany (właściwie to pozbyto się części odpowiedzialnej za komunikację USART). Założono, że wyświetlane będą temperatury z zakresu od 0 do 99 °C. Bufor zawierający próbki temperatur ma 1000 slotów, a kolejne obliczenia temperatur z tysiąca próbek ma miejsce co  $\frac{1}{x}$  sekundy. Triggerem do wyzwalania próbkowania jest Timer 6, którego częstotliwość taktowania ustawiono na 1000 Hz. Domyślnie zmienna  $x$  przyjmuje wartość 1. W przerwaniu od przetwornika analogowo-cyfrowego zbierana jest próbka, która zapisywana jest w buforze cyklicznym. Gdy jest on przeppełniony (w zależności od nadanej wartości zmiennej  $x$ ) wystawiana jest flaga, dzięki której w funkcji głównej obliczana jest nowa temperatura jako średnia arytmetyczna aktualnych próbek zachowanych w buforze. Zależność pozwalającą na odczyt temperatury na podstawie średniej arytmetycznej próbek z bufora przedstawiono poniżej:

$$\frac{10 * (\frac{33*S}{30*N} - T30) * (130 - 30)}{T130 - T30} + 300,$$

gdzie N to rozmiar bufora oznaczony zmienną BUF\_SIZE, S to suma wszystkich próbek zachowanych w buforze, T130 oraz T30 to temperatury odczytane podczas kalibracji czujnika temperatury. Otrzymywane wartości są trzycyfrowe, przy czym trzecia cyfra to dokładność dziesiętna pomiaru.

## 3 Część odpowiedzialna za komunikację I2C

### 3.1 Procedura I2C

Całą procedurę komunikacji z pomocą magistrali I2C można przedstawić w punktach:

1. Włączenie zegara I2C oraz GPIOB z pomocą funkcji I2C\_clock\_enable oraz GPIOB\_clock\_enable.
2. Odpowiednia konfiguracja pinów SCL i SDA w porcie GPIOB - funkcja I2C\_pin\_config.
3. Reset magistrali I2C - funkcja I2C\_reset.
4. Wyłączenie peryferiału, aby dokonać konfiguracji timingr - funkcja I2C\_peripheral\_disable.
5. Konfiguracja timingr (podana przez prowadzącego) - funkcja I2C\_timingr\_config.
6. Włączenie peryferiału - funkcja I2C\_peripheral\_enable.
7. Ustawienie adresu slave'a przesuniętego o jeden w lewo - funkcja I2C\_set\_slave\_address(slave\_address).
8. Rozpoczęcie transmisji.

Większość punktów to pojedyncza zmiana bitów na bazie dokumentacji płytka, ale więcej uwagi wymaga punkt 2. oraz 8.

Podczas konfiguracji pinów w porcie GPIOB ustawiono piny PB8 i PB9 w tryb alternate 4. W tym trybie są one widziane jak odpowiednio linia zegarowa I2C oraz linia danych I2C. Następnie oba piny ustawiono w tryb otwartego drenu, najwyższej częstotliwości oraz pull-up. Rezystory pull-up na linii I2C powodują, że linia jest domyślnie w stanie wysokim 1, a tryb otwartego drenu pozwoli na zmianę tego stanu na 0 w momencie wywołania komunikacji.

### 3.2 Transmisja

Transmisja przebiega w kilku krokach. Pierwszy z nich to ustawienie adresu slave'a jako 0x42 dla dwóch prawych slotów wyświetlacza lub 0x43 - dla dwóch lewych slotów. Są to adresy expanderów, które obsługują po dwa sloty w wyświetlaczu. Drugi krok to konfiguracja, z pomocą funkcji I2C\_7bit\_transfer\_config, liczby wysyłanych bajtów na 3 (rejestr do zapisywania oraz 2 bajty odpowiedzialne za wyświetlanie), wyłączenie reloadu i autoendu oraz ustawienie 7-bitowego trybu write. Następnie włączana jest komunikacja I2C oraz przesyłane są 3 bajty. Na koniec sprawdzenie funkcją I2C\_check\_transfer\_complete czy ukończono transmisję i zakończenie komunikacji.

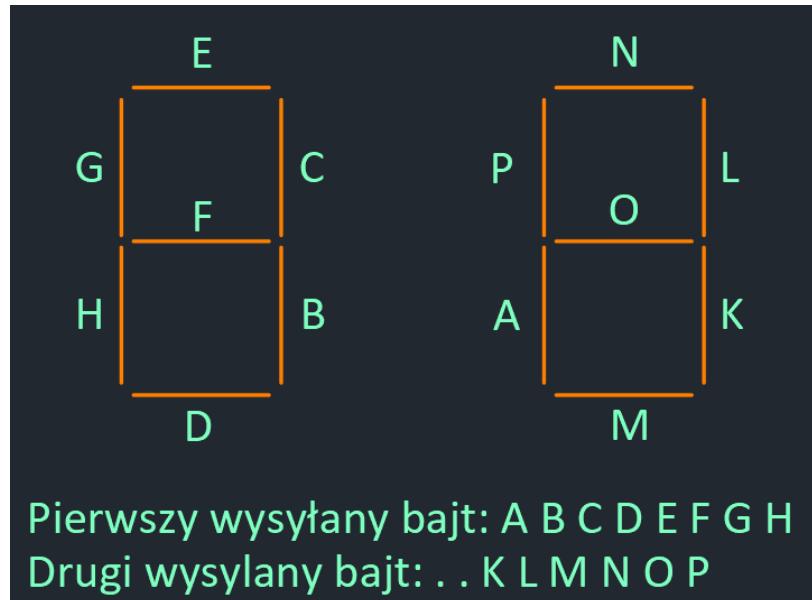
## 4 Funkcja główna

W funkcji głównej zawarto włączenie zegarów, konfigurację I2C, wyświetlenie znaku  $^{\circ}C$ , konfigurację części przetwornika ADC, włączenie obsługi przerwania oraz pętlę główną.

W pętli głównej zawarto część obliczającą temperaturę na podstawie średniej z bufora. Program wykonuje ją, gdy odczyta flagę wystawioną przez przerwanie wymuszającą ponowne obliczenie temperatury, a następnie czyści odpowiednie flagi i iteratory. W dalszej części pętli głównej zawarto część odpowiedzialną za wyświetlenie temperatury. Ponawia ona komunikację I2C na dwóch lewych slotach wyświetlacza tylko w przypadku, gdy obliczona temperatura różni się od poprzedniej. Ten zabieg ma zapobiec wywołaniu nic nie zmieniającej na wyświetlaczu komunikacji I2C.

## 5 Część odpowiedzialna za wyświetlanie liczb

Wykorzystując automatyczne nadpisywanie rejestru odpowiedzialnego za odpowiednie segmenty w slotach realizacja wyświetlania odbywa się poprzez wysłanie kolejno dwóch bajtów wedle schematu z rysunku 2. Z dokumentacji odczytano, że dwa pierwsze bity w drugim bajcie nie są podłączone do żadnego segmentu.

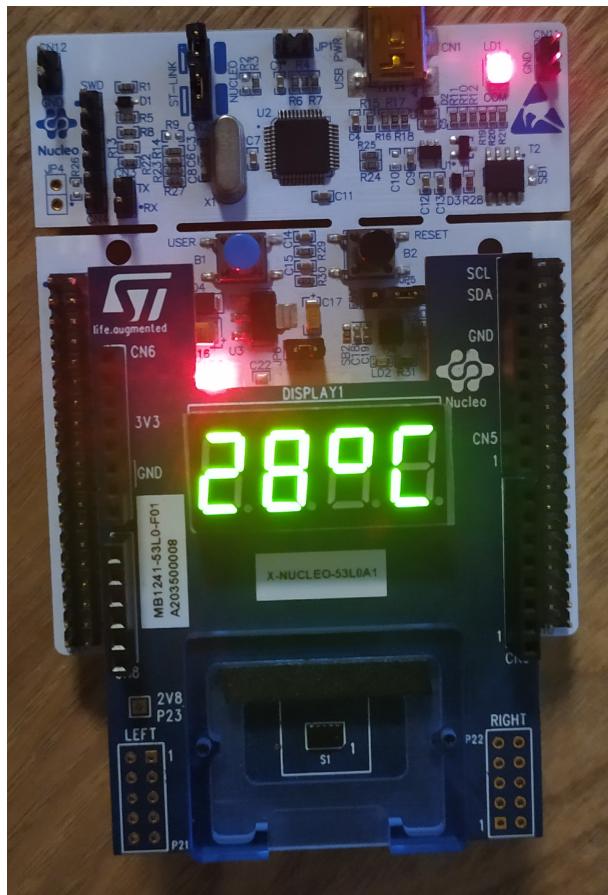


Rysunek 2: Schemat opisujący segmenty obu slotów

Jak widać na powyższym rysunku pierwszy z wysyłanych bajtów obsługuje liczbę dziesiątek oraz jeden segment liczby jedności. Jest to niewielką przeszkodą przy pisaniu funkcji wyświetlających odpowiednie cyfry. Funkcja obsługująca pierwszy slot przyjmuje za argumenty obliczoną w pętli głównej liczbę dziesiątek i jedności, a następnie jeśli liczba jedności jest równa 0, 2, 6 lub 8 do najważniejszego bitu pierwszego bajtu przypisuje 1 (zapala segment). W przeciwnym przypadku przypisuje tam 0 (gasi, bądź nie zapala segmentu). Funkcja odpowiadająca za wyświetlanie drugiego slotu jest już prostsza. Za argument przyjęto liczbę jedności i wedle niej ustalono, które bity należy ustawać. Obie funkcje napisano z wykorzystaniem instrukcji switch, a w przypadku default wyświetla się litera E.

## 6 Zastosowanie

Na zdjęciu 3. przedstawiono efekt działania w postaci wyświetlania aktualnej temperatury mikroprocesora, którą po ustaleniu można utożsamiać z temperaturą otoczenia.



Rysunek 3: Temperatura otoczenia

## 7 Kalibracja wyświetlanej temperatury

W związku z faktem, iż temperatura mierzona jest bezpośrednio na mikrokontrolerze wykonano sprawdzenie rzetelności tego pomiaru. Umieszczono płytę obok termometru i po 45 minutach odczytano wskazania obu termometrów. Termometr wskazywał równo  $24^{\circ}\text{C}$ , a na wyświetlaczu widniało  $26^{\circ}\text{C}$ . Za pomocą śledzenia zmiennej temperatury w kodzie sprawdzono dokładność i temperatura ta wyniosła  $26,2^{\circ}\text{C}$ . Po godzinie przerwy, w tym samym pokoju powtórzono pomiar. Termometr wskazał  $23,5^{\circ}\text{C}$ , a mikrokontroler –  $25,9^{\circ}\text{C}$ .

Kolejnym krokiem był pomiar w wychłodzonym pokoju. Termometr i płytę umieszczono z dala od okna i po 45 minutach odczytano temperatury. Termometr w obu przypadkach wskazał  $16,5^{\circ}\text{C}$ , a mikrokontroler, po ustaleniu wskazał dwukrotnie  $18,5^{\circ}\text{C}$ .

Wywnioskowano, że w chłodniejszym pomieszczeniu różnica temperatur na mikrokontrolerze i powietrza była bardziej stabilna. Wyniosła  $2^{\circ}\text{C}$ . W cieplejszym było to  $2,2^{\circ}\text{C}$  oraz  $2,4^{\circ}\text{C}$ . Postanowiono odjąć od mierzonej temperatury wartość  $2,2^{\circ}\text{C}$ , jako zaokrągloną do wartości dziesiętnej średnią z różnic pomiarów czterech temperatur. Dzięki temu działaniu można postulować, że dokładność pomiaru temperatur pokojowych jest dokładna.