

19.11.2024r.

Sprawozdanie

Internet rzeczy: WEJŚCIA ANALOGOWE, OBSŁUGA WYŚWIETLACZA LCD RGB, MAGISTRALA CAN

1. Cel ćwiczenia - Celem ćwiczenia było zapoznanie z podstawowymi interfejsami komunikacyjnymi używanymi w Arduino UNO.

2. Przebieg ćwiczenia.

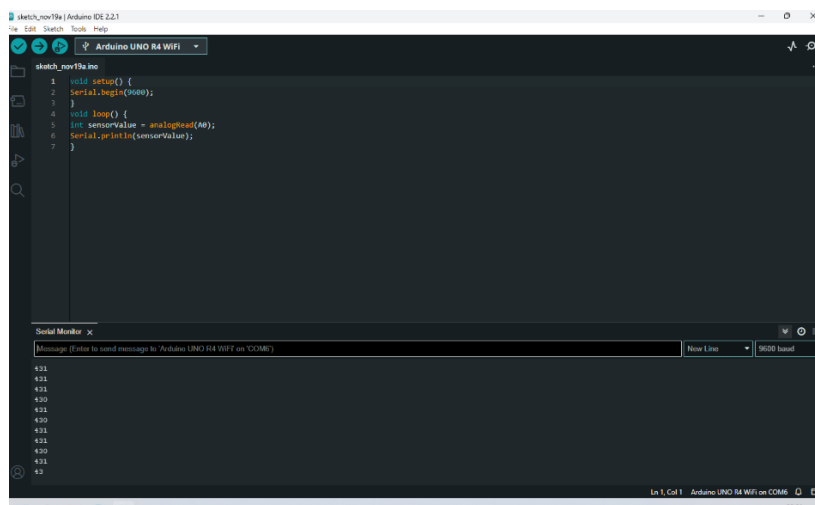
2.1 Pomiar napięcia z wykorzystywaniem wejścia analogowego.

Postępujące zgodnie z instrukcją, wykorzystaliśmy nakładkę Grove Shield w celu podłączenia potencjometru.

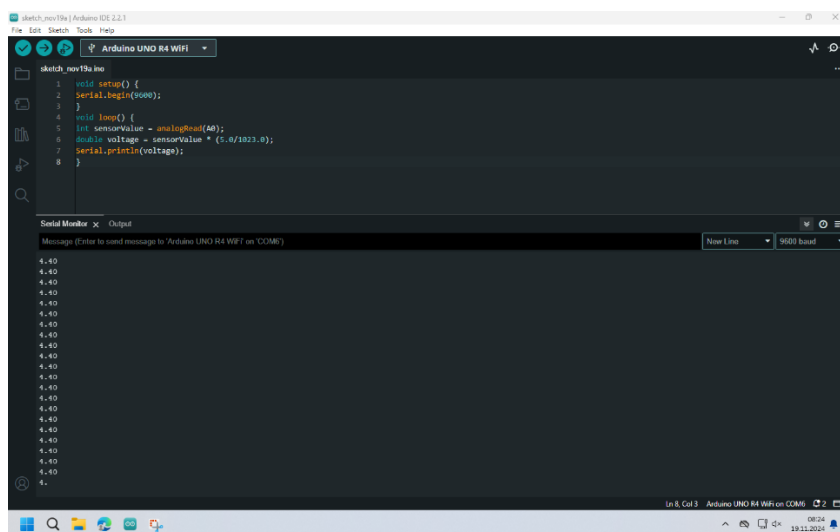


Rys. 2. Podłączenie potencjometru do nakładki Grove

Następnie do Arduino wgraliśmy program odczytujący wartość z wejścia analogowego A0 i wysyłający ją do komputera z wykorzystaniem portu szeregowego. Odczyt był możliwy w Serial Monitor.



Następnie zmodyfikowaliśmy program dodając linię przeliczającą wartości odczytane z przetwornika A/C na napięcie.



Pomiary napięcia były wykonywane do pierwszego miejsca po przecinku.

Po kolejnej modyfikacji kodu, dokładność zmieniła się i pomiary były wykonywane do drugiego miejsca po przecinku.

The screenshot displays the Arduino IDE environment. The top menu bar includes 'File', 'Edit', 'Sketch', 'Tools', and 'Help'. The toolbar contains icons for file operations and execution. The main editor window shows a sketch named 'sketch_nov19a.ino' with the following code:

```

1 void setup() {
2   Serial.begin(9600);
3 }
4 void loop() {
5   int sensorValue = analogRead(A0);
6   double voltage = sensorValue * (5.00/1023.00);
7   Serial.println(voltage);
8 }

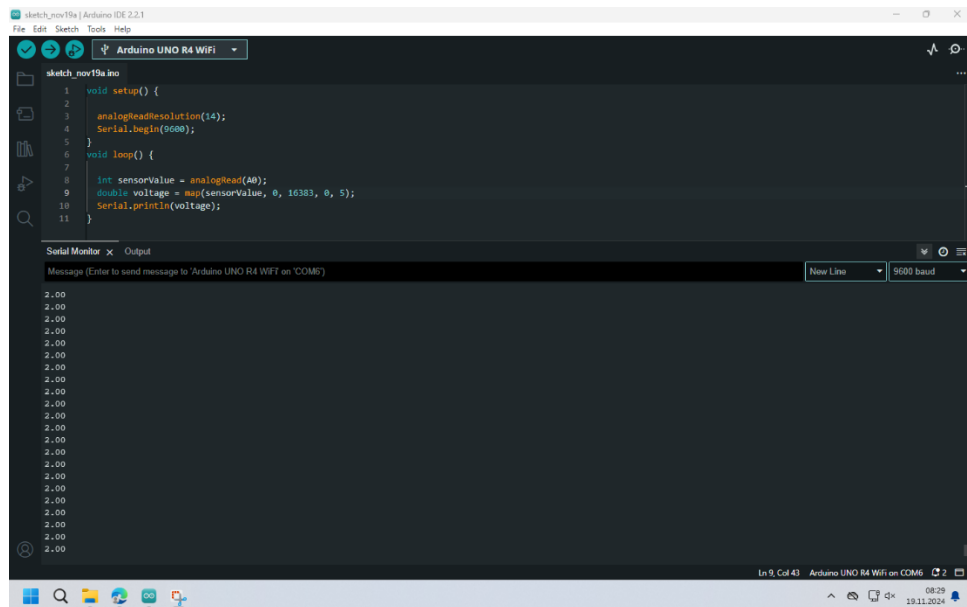
```

Below the editor is the 'Serial Monitor' window, which is currently empty. The status bar at the bottom indicates 'Ln 6, Col 48' and 'Arduino UNO R4 WiFi on COM6'.

Do tej pory pomiary były wykonywane z rozdzielczością 10 bitów. W celu zmiany tego parametru, należało uzupełnić kod o instrukcję „analogReadResolution(14)”, zmieniało to rozdzielczość przetwarzania na 14 bitów.

[illegible]

W celu wykorzystania rozdzielczości 14 bitowej należy użyć w linijce 9 wartości 16383, a w celu wykorzystania 10 bitowej, wartości 1023.



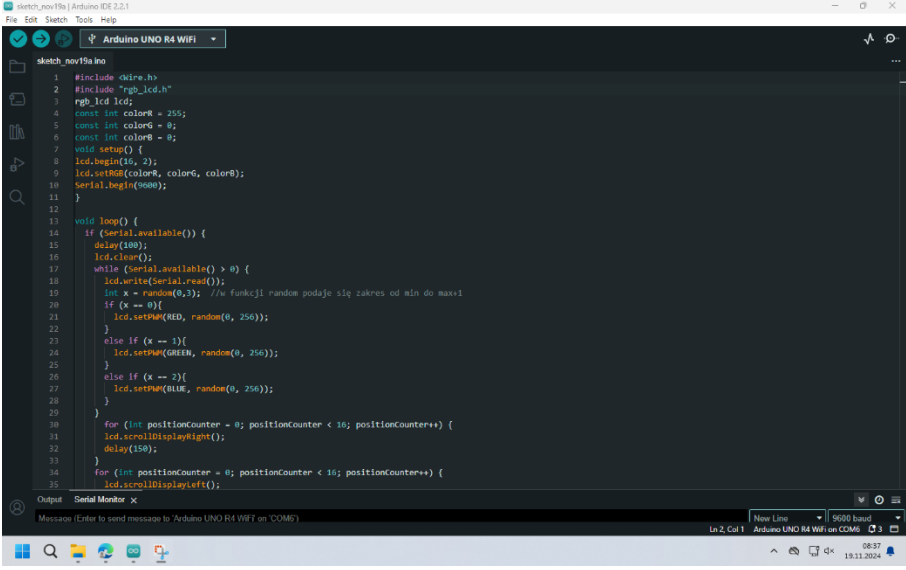
2.2 Obsługa wyświetlacza LCD RGB.

Do wykonania następnej części ćwiczenia użyliśmy podanego wyświetlacza. Wgraliśmy wymaganą do niego bibliotekę oraz podłączyliśmy go według wytycznych.



Rys. 3. Podłączenie wyświetlacza Grove LCD Backlight

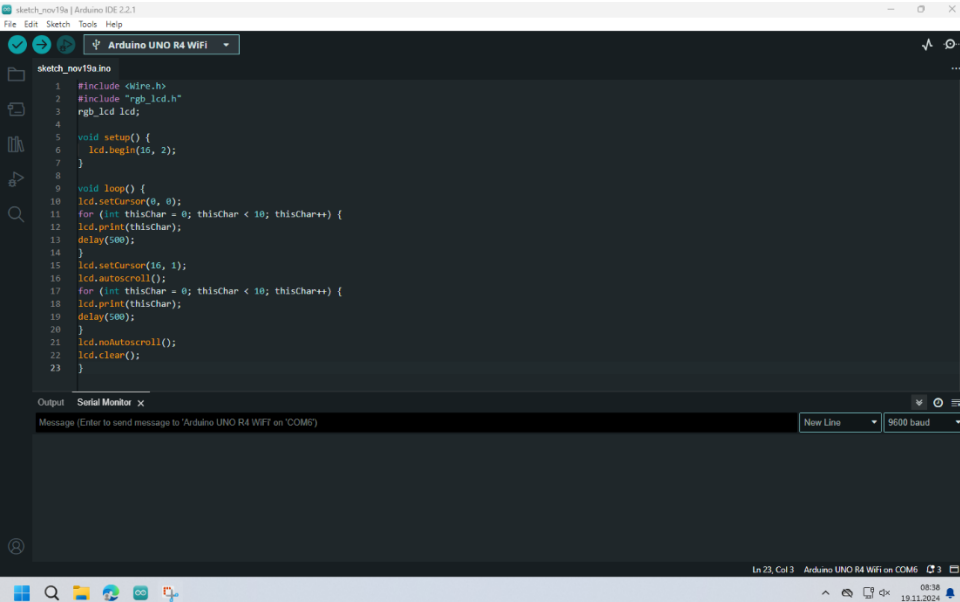
Następnie do Arduino wpisaliśmy program podany w instrukcji, który pokazywał przykładowe możliwości wyświetlacza.



```
1 #include <Wire.h>
2 #include "rgb_lcd.h"
3 rgb_lcd lcd;
4 const int colorR = 255;
5 const int colorG = 0;
6 const int colorB = 0;
7 void setup() {
8   lcd.begin(16, 2);
9   lcd.setRGB(colorR, colorG, colorB);
10  Serial.begin(9600);
11 }
12
13 void loop() {
14   if (Serial.available()) {
15     delay(100);
16     lcd.clear();
17     while (Serial.available() > 0) {
18       lcd.write(Serial.read());
19       int x = random(0,2); //x funkcji random podaje się zakres od min do max+1
20       if (x == 0){
21         lcd.setRGB(RED, random(0, 256));
22       }
23       else if (x == 1){
24         lcd.setRGB(GREEN, random(0, 256));
25       }
26       else if (x == 2){
27         lcd.setRGB(BLUE, random(0, 256));
28       }
29     }
30     for (int positionCounter = 0; positionCounter < 16; positionCounter++) {
31       lcd.scrollDisplayRight();
32       delay(150);
33     }
34     for (int positionCounter = 0; positionCounter < 16; positionCounter++) {
35       lcd.scrollDisplayLeft();
36     }
37   }
38 }
```

Do obsługi tego wyświetlacza są potrzebne biblioteki Wire oraz rgb_lcd, pierwsza odpowiada za komunikację poprzez magistralę I2C, natomiast biblioteka rgb_lcd odpowiada za możliwość interakcji z wyświetlaczem.

Kolejnym krokiem było przetestowanie programu pokazującego sterowanie kursorem na wyświetlaczu.



```
1 #include <Wire.h>
2 #include "rgb_lcd.h"
3 rgb_lcd lcd;
4
5 void setup() {
6   lcd.begin(16, 2);
7 }
8
9 void loop() {
10  lcd.setCursor(0, 0);
11  for (int thisChar = 0; thisChar < 10; thisChar++) {
12    lcd.print(thisChar);
13    delay(500);
14  }
15  lcd.setCursor(15, 1);
16  lcd.autoscroll();
17  for (int thisChar = 0; thisChar < 10; thisChar++) {
18    lcd.print(thisChar);
19    delay(500);
20  }
21  lcd.noAutoscroll();
22  lcd.clear();
23 }
```

Funkcja setCursor pozwala na ustawienie kursora w danym miejscu na wyświetlaczu, parametry odpowiadają wierszowi i kolumnie na wyświetlaczu.

Zadania 3 i 4 nie udało się wykonać z powodu problemu z magistralą CAN, moduły posiadały niepodłączone rezystory PULLUP, które uniemożliwiały wykonanie tych zadań.

3. Wnioski

- Wprowadzenie większej precyzji w zapisie współczynnika przeliczeniowego, np. 5.00/1023.00 pozwoliło na dokładniejsze wyznaczenie wartości napięcia, zmniejszając zaokrąglenia w obliczeniach.
- Przy zastosowaniu przetwornika o wyższej rozdzielczości (np. 14 bitów), uzyskano by bardziej szczegółowe wyniki pomiarów. Kod można rozszerzyć o dynamiczne skalowanie napięcia przy różnych zakresach zasilania.
- Komunikacja z wyświetlaczem odbywa się przez magistralę I²C z wykorzystaniem biblioteki Wire oraz dedykowanej biblioteki rgb_lcd. Pozwala to na łatwe sterowanie treścią wyświetlaną na ekranie i kolorami podświetlenia.
- Kod demonstruje funkcję scrollowania tekstu na wyświetlaczu oraz jego dynamiczną aktualizację. Przy odpowiednim sterowaniu można wyświetlać nie tylko tekst, ale również dane pomiarowe (np. wyniki z zadania 1).