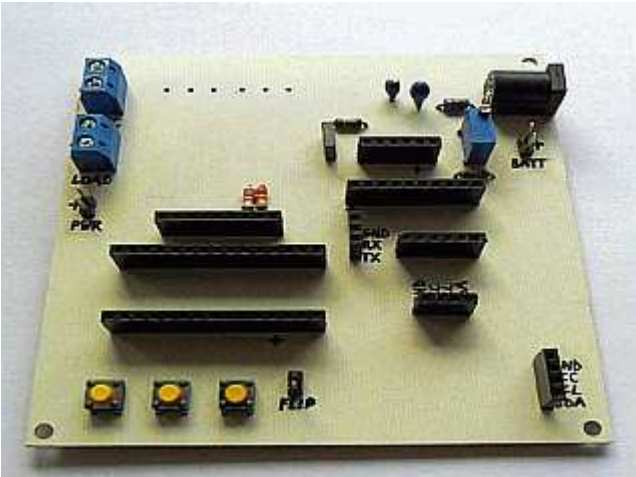


# ArdDis 1.04

© 2025 by RomanWorkshop

<http://romanworkshop.blutu.pl/>

# Rejestrator rozładowywania (Arduino)



Urządzenie służy do monitorowania i rejestrowania procesu rozładowywania dowolnego źródła (np. akumulatora) o napięciu  $V_{in}$  do 40V, które podłącza się do złącza CON3 (Source). Maksymalny prąd rozładowywania nie powinien przekroczyć ok. 3A. Konieczne jest użycie zewnętrznego obciążenia stałoprądowego, podłączanego do złącza CON4 (Load). Napięcie zasilania DC dla obciążenia jest dostępne na złączu CON5 (Power). Wartość napięcia  $V_{in}$  z badanego źródła jest cały czas mierzona i zapisywana na karcie SD do pliku CSV w formacie programu "TsDMMViewer", który pozwala uzyskiwać dowolnie skalowalne wykresy. Pomiary trwają do momentu, gdy zmierzona wartość napięcia Volt ze źródła, spadnie poniżej ustawionej

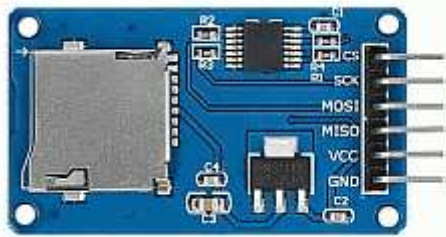



wartości napięcia odcięcia  $V_{off}$  (0.1-20V). Wtedy źródło jest automatycznie odłączane od obciążenia. Wszystkie informacje są prezentowane na wyświetlaczu OLED oraz wysyłane przez interfejs szeregowy USART (złącze CON6). Do obsługi urządzenia służą trzy przyciski/mikrostryki: S1 (Start/Stop), S2 ( $V_{off}$ -) i S3 ( $V_{off}$ +).

## 1. Wstęp

Rejestrator rozładowywania jest zbudowany z kilku łatwo dostępnych i tanich modułów:

NAZWA	WYGLĄD	OPIS
MOD1		Arduino Nano 3.0 z mikrokontrolerem ATmega328, taktowanym zegarem o częstotliwości 16 MHz.
MOD2		16-bitowy przetwornik analogowo-cyfrowy ADC ADS1115 (adres I2C 0x48).
MOD3		zegar czasu rzeczywistego RTC DS3231 (adres I2C 0x68) z opcjonalną szeregową pamięcią EEPROM 24C32 (adres I2C 0x57).

Wymaga przeróbki - patrz opis.

MOD4		adapter ze złączem kart microSD (SPI).
MOD5		2 przekaźniki 5V (JQC-3FF-S-Z). Wymaga przeróbki - patrz opis.
MOD6A		opcjonalny wyświetlacz OLED o rozdzielczości 128x64/128x32 pikseli ze sterownikiem SSD1306 (adres I2C 0x3D/0x3C).
MOD6B		

Wszystkie moduły posiadają wlutowane listwy kołkowe goldpin (kątowe), które są wpinane do odpowiednich żeńskich złączy goldpin na płytce urządzenia. Taka budowa pozwala na łatwą wymianę każdego modułu osobno.

Zworka J1 służy do obracania (flip) o 180 stopni wyświetlanego na OLED obrazu. Przy zdjętej zworce (OFF) obraz jest obrócony (po resecie/włączeniu zasilania). Wyświetlacze OLED mają różne wymiary, rozmieszczenie otworów montażowych oraz sygnałów na wyprowadzeniach modułu. Płytkę zawiera obrys największego OLED jaki miałem bez otworów oraz sygnały w kolejności: GND, VCC, SCL, SDA. Pod wyświetlacz można przykleić jakiś plastikowy dystans, aby nie opadał po drugiej stronie złącza goldpin, ale nie jest to konieczne (złącze trzyma go dość stabilnie).

Program sterujący kompilowałem w środowisku "Arduino 1.8.19" z zainstalowanymi bibliotekami: "ADS1115\_WE 1.5.0", "DS3231 1.1.2", "ss\_oled 4.3.1" + "BitBang\_I2C 2.2.1", "JC\_EEPROM 1.0.8". W archiwum oprócz wszystkich szkiców, znajdują się wsady programu sterującego "ArdDis.hex" oraz programu do sprawdzania modelu przetwornika ADC "ADS1115test.hex". Wsady są też dostępne w wersji z dołączonym nowym/starym bootloaderem Arduino: "\_new\_bootloader.hex" / "\_old\_bootloader.hex".

Urządzenie można zasiląć napięciem stałym DC=7-12V przez złącze CON2 (DC-Jack, plus w środku wtyczki) lub CON1 (goldpin) z zasilacza lub z baterii/akumulatorów, ale nie z tych dwóch źródeł jednocześnie. Dioda D1 zabezpiecza przed odwrotną polaryzacją napięcia zasilania DC (występuje na niej spadek 300mV przy prądzie do 100mA). Napięcie zasilania DC na złączu CON1/CON2, należy wyłączyć przed podłączeniem modułu Arduino Nano do portu USB komputera/laptopa.

Modelowy rejestrator z wyświetlaczem OLED 128x64 przy napięciu DC=9V, pobierał prąd ok. 185mA w stanie spoczynku (z wyłączonymi przekaźnikami ok. 36mA). Zmontowałem go na płytce jednostron-

nej o wymiarach 100x100mm (bez zworek), wykonanej metodą transferu chemicznego. W rogach płytki znajdują się cztery otwory montażowe o średnicy 3mm, dzięki którym urządzenie można przykręcić do obudowy lub zamontować w nich jakieś nóżki w przypadku jej braku.

## 2. Uruchomienie

Konieczne jest wykonanie przeróbek sprzętowych dwóch modułów. Moduł zegara DS3231 posiada koszyk na baterię/akumulator 3V 2032 do podtrzymywania czasu. Dodatkowo zawiera też układ ładujący (rezystor 1k i dioda 1N4148) przeznaczony dla akumulatora LIR2032. Ładowanie działa tylko wtedy, gdy moduł jest zasilany przez pin VCC. Jeśli korzystamy ze zwykłej baterii CR2032, należy odłączyć układ ładujący przez wylutowanie rezystora lub diody (można też przeciąć ścieżkę tuż przy katodzie diody, która biegnie do plusa w koszyku). Zaniechanie tej przeróbki może spowodować skrócenie życia baterii, a nawet jej spuchnięcie i wylanie.

W module z 2 przekaźnikami 5V (JQC-3FF-S-Z), należy wylutować kołki goldpin 3x1/4x1 i wlutować je z drugiej strony płytki modułu tak, aby weszły do złącza MOD5 na płytce urządzenia. Zamiast dwóch żeńskich złączy goldpin 3x1/4x1, wlutowałem jedno 9x1 z wyjętymi stykami nr 4 i 5. Z modułu trzeba też wylutować 2 gniazda śrubowe ARK 3-pin, a w miejscu ich styków COM1, NC1, COM2 i NC2 (styki NO1 i NO2 nie są podłączone) wlutować pionowo 4 druty miedziane o średnicy 1.1-1.2mm i długości 17mm, które będą łączyły moduł z płytką urządzenia. Zamiast 4 drutów można by wlutować 2 listwy kołkowe goldpin 3x1 z wyjętym środkowym kołkiem, a w płytkę urządzenia 2 żeńskie złącza goldpin 3x1 z wyjętym środkowym stykiem. Jednak takie rozwiązanie zmniejsza maksymalny prąd obciążenia, który wynosi ok. 2A dla pojedynczego kołka goldpin oraz ok. 1A dla żeńskiego styku goldpin. W przypadku gorszej jakości (np. chińskich) złączy goldpin, lepiej nie przekraczać prądu o wartości kilkuset mA. Dla miedzianego drutu o średnicy 1.1-1.2mm maksymalny prąd obciążenia wynosi ok. 2.7-3.2A.

Przy wkładaniu każdego modułu do jego złącza, należy bardzo uważać aby nie włożyć go odwrotnie. Po zamontowaniu wszystkich niezbędnych modułów można sprawdzić działanie przetwornika ADC. W tym celu, należy wyłączyć napięcie zasilania DC na złączu CON1/CON2 i podłączyć moduł Arduino Nano do portu USB komputera/laptopa.

Niektóre moduły oznaczone jako "ADS1115" w rzeczywistości mogą zawierać przetwornik ADS1015 (12-bit), a moduły oznaczone "ADS1015" mogą mieć przetwornik ADS1115 (16-bit). Teoretycznie oba układy można rozróżnić po oznaczeniu nadrukowanym na ich obudowie: "BRPI" = ADS1015, "BOGI" = ADS1115. Niestety zdarzają się przypadki fałszowania/przemałowywania tych oznaczeń. W takiej sytuacji jedynym pewnym sposobem sprawdzenia modelu przetwornika jest uruchomienie szkicu "ADS1115test.ino" lub wgranie wsadu "ADS1115test.hex". Określa on model przetwornika na podstawie: wartości 4 najmłodszych bitów w wynikach 10 konwersji (dla ADS1015 zawsze mają wartość 0) oraz czasu wykonania 10 konwersji (dla ADS1015 zawsze jest krótszy). Wgrywamy program do urządzenia i uruchamiamy je. Na USART/OLED pojawi się komunikat "Connect Vin to CON3 and press S1". Wtedy do złącza CON3 podłączamy dowolne źródło o napięciu Vin=1-40V i naciskamy przycisk S1. Po krótkiej chwili na USART/OLED pojawi się komunikat z wynikiem testu: "ADS1015 detected" / "ADS1115 detected" - jeśli wykryto ADS1015/ADS1115 lub "Undetermined result" - jeśli wynik nie jest jednoznaczny. W ostatnim przypadku zaleca się zwiększenie wartości podłączonego napięcia Vin i powtórzenie testu. Sprawdzanie modelu przetwornika, należy przeprowadzać bez obciążenia podłączonego do złącza CON4.

Kolejnym krokiem jest ustawienie aktualnej daty i czasu zegara DS3231. Można to zrobić podając w szkicu "DS3231set.ino" aktualną datę (dzień, miesiąc, rok), numer dnia tygodnia (1 = poniedziałek), tryb pracy zegara (0 = 24h) oraz dokładny czas (sekunda, minuta, godzina) z wyprzedzeniem, np. 2 minut. Następnie wgrywamy program do urządzenia i uruchamiamy je. Na USART/OLED pojawi się komunikat "Wait for press S1". Kiedy na zegarze wzorcowym pojawi się podany w szkicu czas, naciskamy przycisk S1. Podany czas zostanie zapisany w zegarze DS3231, a na USART/OLED pojawi się komunikat "Date/Time SET". Od tego momentu na USART/OLED będzie ciągle

wyświetlany (ok. 4x/s) ciąg "dd-mm-yyyy hh:ii:ss" z aktualną datą i czasem, które są odczytywane z zegara DS3231. Dodatkowo na USART za tym ciągiem zostanie też wyświetlony numer dnia tygodnia. Niektóre moduły mogą zawierać układ DS3231M, który ma nieco gorszą stabilność (temperatura pracy -45-85C, stabilność 5ppm = 0.43s/dzień), niż układ DS3231S/DS3231SN (temperatura pracy 0-70C/-40-85C, stabilność 3.5ppm = 0.3s/dzień, przy temperaturze 0-40C typowo 2ppm = 0.17s/dzień).

Następnie wgrywamy do urządzenia program sterujący, poprzez uruchomienie szkicu "ArdDis.ino" lub wgranie wsadu "ArdDis.hex". Moduł Arduino Nano może mieć fabrycznie wgrany nowy ("ATmega328P") lub stary ("ATmega328P (Old Bootloader)") bootloader. W razie problemów z wgraniem szkicu, należy w środowisku Arduino zmienić bootloader na przeciwny i spróbować ponownie. Na czas programowania może być konieczne wyjęcie modułu Arduino Nano z urządzenia, a później jego włożenie. Po wgraniu programu sterującego, odłączamy moduł Arduino Nano od portu USB komputera/laptopa i włączamy napięcie zasilania DC na złączu CON1/CON2.

Ostatnim etapem jest kalibracja wskazania pomiaru napięcia, używając dowolnego źródła o napięciu  $V_{in}=5-15V$  oraz fabrycznego multimetru. W tym celu podczas resetu/włączania urządzenia, należy przytrzymać wciśnięty przycisk S2. Program przejdzie w tryb kalibracji (jego włączenie nie wymaga obecności karty SD), czyli ciągłego pomiaru napięcia  $V_{in}$  podanego na złącze CON3. Zmierzona wartość napięcia Volt ze źródła wraz z aktualną datą/godziną odczytaną z zegara DS3231, będzie ciągle wyświetlana (ok. 4x/s) na USART/OLED. Do napięcia  $V_{in}$  podłączamy jednocześnie fabryczny multimetr, ustawiony na zakres pomiaru napięcia stałego do 20V. Regulujemy potencjometrem P1, aż do uzyskania identycznego wskazania jak na multimetrze. Dzięki temu trybowi można sprawdzić dokładność pomiaru napięcia oraz aktualność czasu zegara DS3231. Aby wyjść z trybu kalibracji, należy nacisnąć przycisk S3. Kalibrowanie wskazań napięcia, należy przeprowadzać bez obciążenia podłączonego do złącza CON4.

### 3. Działanie

Po włączeniu zasilania przekaźniki K1/K2 zostają włączone niskim stanem logicznym na wejściach IN1/IN2, który jest wymuszony przez rezystory R3/R4 podłączone do masy. W stanie załączenia (zwarte styki COMx-NOx) przekaźniki pobierają ok. 65mA prądu każdy. Podczas pomiarów przekaźniki są wyłączone (zwarte styki COMx-NCx) i nie pobierają prądu, co znacznie ogranicza zużycie energii.

Program sterujący komunikuje się z użytkownikiem, wysyłając dane tekstowe przez interfejs szeregowy USART (złącze CON6) z szybkością 9600 B/s w trybie 8N1 (8-bitów danych, bez bitu parzystości, 1-bit stopu). Jeśli na magistrali I2C zostanie wykryty wyświetlacz OLED 128x32/128x64 (sterownik SSD1306) o adresie 0x3C (0x78)/0x3D (0x7A), to również na nim będą wyświetlane informacje. Zworka J1 służy do obracania (flip) o 180 stopni wyświetlanego obrazu. Przy zdjętej zworce (OFF) obraz jest obrócony (po resecie/włączeniu zasilania). Na wyświetlaczu o rozdzielczości 128x64 obraz jest wyświetlany z 1-liniowymi poziomymi przerwami (co druga pozioma linia jest pusta, przez co obraz jest rozciągnięty w pionie). Jeśli wyświetlacz OLED nie zostanie wykryty, to na USART pojawi się komunikat "SSD1306 missing".

Następnie program sprawdza, czy na magistrali I2C są obecne następujące moduły: 16-bitowy przetwornik analogowo-cyfrowy ADS1115 (0x48 - pin ADDR podłączony przez rezystor 10k do masy) i zegar czasu rzeczywistego DS3231 (0x68). Są one niezbędne do działania urządzenia więc jeśli któregoś brakuje, to na USART/OLED pojawi się komunikat "ADS1115 missing" lub "DS3231 missing", a program zostanie zatrzymany.

Jeśli w tym miejscu program wykryje naciśnięty przycisk S2, to włączy się tryb kalibracji. Zostaje wyłączony przekaźnik K1 (zwarte styki COM1-NC1), co powoduje podłączenie badanego źródła do dzielnika napięcia 10:1 i obciążenia stałoprądowego. Po odczekaniu 200ms (stabilizacja drgań styków przekaźnika i wartości napięcia ze źródła), będą wykonywane ciągłe pomiary napięcia  $V_{in}$  po-



danego na złącze CON3. Zmierzona wartość napięcia Volt ze źródła wraz z aktualną datą/godziną odczytaną z zegara DS3231, będzie ciągle wyświetlana (ok. 4x/s) na USART/OLED. Tryb ten służy do sprawdzania dokładności pomiaru napięcia oraz aktualności czasu zegara DS3231. Aby wyjść z trybu kalibracji, należy nacisnąć przycisk S3. Wtedy przełącznik K1 zostaje włączony (rozwarło styki COM1-NC1), co powoduje odłączenie badanego źródła od dzielnika napięcia 10:1 i obciążenia stałoprądowego. Kalibrowanie wskazań napięcia, należy przeprowadzać bez obciążenia podłączonego do złącza CON4.

Moduł zegara DS3231 może posiadać wlutowany układ szeregowej pamięci EEPROM 24C32 (0x57). Jeśli program wykryje ją na magistrali I2C, to będzie ona używana do zapamiętania ostatnio ustawionej wartości napięcia odcięcia Voff. Przed wykonywaniem kolejnych pomiarów, wartość Voff będzie odczytywana z pamięci i wyświetlana na USART/OLED zamiast wartości domyślnej. Jeśli pamięć nie zostanie wykryta lub wystąpi błąd jej odczytu, to na USART pojawi się komunikat "EEPROM missing", a przed wykonywaniem kolejnych pomiarów na USART/OLED, będzie wyświetlana domyślna wartość Voff=10V.

Następnie program sprawdza, czy na magistrali SPI znajduje się karta SD w formacie FAT16/FAT32, włożona do modułu adaptera ze złączem kart microSD, stabilizatorem LDO 1117-3.3V oraz translatorem poziomu stanów logicznych 5/3.3V 74LVC125. Jest ona niezbędna do działania urządzenia, więc jeśli jej brakuje lub posiada system plików inny niż FAT16/FAT32, to na USART/OLED pojawi się komunikat "SD card missing", a program zostanie zatrzymany.

Kolejnym etapem programu jest ustawianie wartości napięcia odcięcia Voff w zakresie 0.1-20V, za pomocą przycisków S1-S3. Na USART/OLED będzie ciągle wyświetlana (ok. 10x/s) aktualna wartość napięcia Voff. Jeśli pamięć EEPROM została wykryta będzie to wartość z niej odczytana. W przeciwnym razie będzie to domyślna wartość Voff=10V. Aby zmniejszyć/zwiększyć wartość Voff o 0.1V, należy krótko (poniżej 100ms) nacisnąć przycisk S2/S3. Naciśnięcie i dłuższe przytrzymanie wciśniętego przycisku S2/S3, spowoduje ciągłe zmniejszanie/zwiększanie wartości Voff z szybkością ok. 1V/s, aż do osiągnięcia wartości granicznej, czyli odpowiednio 0.1V/20V. Aby zatwierdzić aktualnie ustawioną wartość napięcia Voff, należy krótko (poniżej 100ms) nacisnąć przycisk S1. Jeśli pamięć EEPROM jest dostępna, to zostanie w niej zapisana wartość Voff. Zapis odbędzie się tylko wtedy, gdy aktualna wartość Voff jest inna od tej, która już znajduje się w komórce pamięci (ograniczenie zużycia pamięci).

Po wybraniu wartości napięcia Voff, program generuje nazwę kolejnego pliku z wynikami pomiarów, który zostanie zapisany w głównym katalogu karty SD. Nazwa ta ma format "LOGxx.CSV", gdzie xx=00-99. Program wyszukuje pierwszą dostępną nazwę poczynawszy od "LOG00.CSV", sprawdzając czy plik o takiej nazwie już istnieje. Jeśli tak, to jest generowana i sprawdzana kolejna nazwa ("LOG01.CSV", "LOG02.CSV"...), aż do znalezienia jeszcze nie używanej. Jeśli wszystkie nazwy w zakresie od "LOG00.CSV" do "LOG99.CSV" są zajęte, to wyniki pomiarów zostaną nadpisane w ostatnim pliku o nazwie "LOG99.CSV".

Następnie program otwiera do zapisu plik o wygenerowanej nazwie, który będzie zawierał wyniki pomiarów (data jego utworzenia pochodzi z zegara DS3231). Jest to plik tekstowy CSV (Comma-Separated Values) z wartościami oddzielonymi przecinkiem, który można otworzyć w programie "TsDMMViewer". Program ten pozwala uzyskiwać bardzo ładne i dowolnie skalowalne wykresy, przedstawiające w czytelny i przejrzysty sposób proces rozładowywania. Otwarcie pliku jest niezbędne do działania urządzenia więc jeśli się to nie uda, bo np. ostatni plik "LOG99.CSV" ma ustawiony atrybut tylko-do-odczytu (Read-Only), to na USART/OLED pojawi się komunikat "Cant open: LOGxx.CSV", a program zostanie zatrzymany. Po otwarciu pliku na USART/OLED pojawi się komunikat "File: LOGxx.CSV". Na USART zostanie wysłana i zapisana do pliku pierwsza linia nagłówkowa "[TsDigitalMultiMeterViewer Ver.9.9.9 : [ArdDis] : Date dd-mm-yyyy hh:ii:ss],LOGxx.CSV (I=\_A Voff=x.xxV),ArdDis". Zawiera ona aktualną datę/czas oraz nazwę wykresu składającą się z: nazwy pliku, wartości użytego prądu rozładowującego (trzeba ją wpisać ręcznie w pliku wynikowym) i ustawionej wartości napięcia odcięcia Voff. Później na USART zostanie wysłana i zapisana do pliku druga linia nagłówkowa " Elapsed time | Main Value | Mode | Date Time".

Następnie zostaje wyłączony przełącznik K1 (zwarte styki COM1-NC1), co powoduje podłączenie badanego źródła do dzielnika napięcia 10:1 i obciążenia stałoprądowego. W tym momencie użyte obciążenie podłączone do złącza CON4, nie powinno jeszcze działać i obciążać badanego źródła. Po odczekaniu 200ms (stabilizacja drgań styków przełącznika i wartości napięcia ze źródła) jest zapisywany znacznik aktualnego czasu w formacie UNIX, który określa czas rozpoczęcia pomiarów. Program przechodzi do wykonywania głównej pętli, w której są wykonywane ciągłe pomiary napięcia Volt z badanego źródła. Dane na wyświetlaczu OLED są aktualizowane co 1s i zajmują 4 linie: "Volt: xx.xx V" - aktualna wartość napięcia Volt, "Voff: xx.xx V" - ustawiona wartość napięcia Voff, "Time: hh:ii:ss" - czas trwania pomiarów, "File: LOGxx.CSV" - nazwa pliku wynikowego. Natomiast co 2s jest wykonywany pomiar rejestrowany, który jest wysyłany na USART i zapisywany do pliku w osobnej linii " hh:ii:ss", DC , xx.xx , V , , dd-mm-yyyy hh:ii:ss,". Linia ta zawiera następujące wartości oddzielone przecinkami: sztuczna oś czasu zwiększana co 2s, zakres pomiarowy multimetru (DC - napięcie stałe), aktualna wartość napięcia Volt, jednostka pomiarowa (V - wolty), tryb (puste), data i godzina pomiaru. Pomiary są wykonywane ciągle, aż do wystąpienia jednego z warunków ich przerwania/zakończenia. Po wykonaniu 2 pomiarów rejestrowanych, zostaje wyłączony przełącznik K2 (zwarte styki COM2-NC2), co powoduje podłączenie napięcia zasilania DC do obciążenia stałoprądowego, które pojawia się na złączu CON5. Jest to napięcie DC ze złącza CON1/CON2, którym jest zasilane urządzenie.

Od tego momentu badane źródło jest obciążone (dzięki takiemu opóźnieniu można zauważyć spadek napięcia źródła pod obciążeniem). Aby przerwać pomiary w dowolnej chwili, należy długo (do 2s) przytrzymać wciśnięty przycisk S1. Pomiary zostaną zakończone automatycznie, jeśli zmierzona napięcie Volt z badanego źródła jest niższe od ustawionego napięcia odcięcia Voff. Po przerwaniu/zakończeniu pomiarów zostają włączone przełączniki K1/K2 (rozwarne styki COM1-NC1/COM2-NC2), co powoduje odłączenie badanego źródła od dzielnika napięcia 10:1 i obciążenia stałoprądowego oraz odłączenie napięcia zasilania DC od obciążenia. Po odczekaniu 200ms (stabilizacja drgań styków przełącznika) jest wykonywany ostatni rejestrowany pomiar napięcia o wartości 0.0V, który jest potrzebny do wygenerowania kompletnego (zamkniętego) wykresu przez program "TsDMMViewer". Na wyświetlaczu OLED będzie widniała wartość napięcia Volt, zmierzona przed odłączeniem badanego źródła. Następnie program zamyka otwarty wcześniej plik z wynikami pomiarów. Na USART pojawi się komunikat "Stop: LOGxx.CSV (xxxx samples)" z liczbą zapisanych pomiarów (próbek). Na wyświetlaczu OLED w ostatniej linii pojawi się komunikat "Stop: LOGxx.CSV", a program zostanie zatrzymany. Jeśli podczas pomiarów zasilanie urządzenia zostanie wyłączone, to wszystkie dane zapisane w pliku wynikowym zostaną utracone (plik na karcie SD będzie pusty). Pomiary można wznowić resetując moduł Arduino Nano przez naciśnięcie umieszczonego na nim przycisku lub wyłączając i włączając zasilanie urządzenia.

## 4. Wady

Pomiary nie są rejestrowane dokładnie co 2s, bo ich interwał wyznacza opóźnienie programowe (w czasie pomiarów trwających 5:05:34, wygenerowana sztuczna oś czasu była opóźniona o 30s). Rozwiązaniem byłoby użycie częstotliwości 1Hz z wyjścia INT/SQW modułu zegara DS3231 do precyzyjnego odmierzenia interwału wykonywania pomiarów.

Przy prądzie obciążenia 2A rejestrowane napięcie, występujące na dzielniku pomiarowym 10:1 jest o ok. 100mV niższe, niż mierzone bezpośrednio na biegunach badanego akumulatora 3.7V (spadek napięcia na rezystancji wszystkich połączeń). Rozwiązaniem byłoby przerwanie ścieżki VIN do rezystora R1 i użycie dodatkowego przewodu, który łączyłby go bezpośrednio z dodatnim (+) biegunem badanego źródła (pomiar napięcia z pominięciem ścieżki prądowej). Przed pomiarami lub podczas kalibracji wskazań napięcia, trzeba by było ręcznie podłączyć źródło do dzielnika za pomocą tego przewodu. Dzielnik byłby ciągle podłączony do źródła i obciążał je rezystancją ok. 100k (również po zakończeniu pomiarów), aż do ręcznego odłączenia przewodu. Program sterujący musiałby generować stałą wartość 0.0V (bez mierzenia) dla ostatniego rejestrowanego pomiaru napięcia, która jest zapisywana w pliku wynikowym i niezbędna do utworzenia kompletnego (zamkniętego) wykresu przez program "TsDMMViewer".

## 5. Poprawki

W archiwum w katalogu "ArdDis\_1.1", znajduje się szkic i wsady programu sterującego dla rejestratora w wersji 1.1. Poprawiłem w nim opisane wcześniej wady, ale wymaga to wykonania modyfikacji na płycie drukowanej:

Pomiary rejestrowane dokładnie co 2s - trzeba połączyć przewodem wyjście INT/SQW modułu zegara DS3231 z pinem D4 (PD4) modułu Arduino Nano. Na wyjściu INT/SQW (z otwartym drenem) jest generowana częstotliwość 1Hz sygnalizująca upływ 1s (pin D4 ma włączony wewnętrzny rezystor podciągający). Każdy pomiar jest synchronizowany tą częstotliwością.

Eliminacja spadku mierzonego napięcia, występującego przy większym prądzie obciążenia - trzeba przeciąć ścieżkę VIN biegnącą do rezystora R1 i połączyć go przewodem, bezpośrednio z dodatnim (+) biegunem badanego źródła (pomiar napięcia z pominięciem ścieżki prądowej). Ponieważ po zakończeniu pomiarów, podłączony w ten sposób dzielnik 1:10 nie jest automatycznie odłączany od źródła, program sterujący generuje stałą wartość 0.0V (bez mierzenia) dla ostatniego rejestrowanego pomiaru napięcia (niezbędną do utworzenia kompletnego/zamkniętego wykresu przez program "TsDMMViewer").

Dużo nowych modułów wyświetlaczy OLED 128x32/128x64 nie działało z rejestratorem. Przyczyną tego problemu był algorytm wykrywania typu sterownika wyświetlacza (SSD1306/SH1106/SH1107) w bibliotece "ss\_oled 4.3.1", który nie obsługiwał niektórych wartości identyfikacyjnych, odczytywanych z rejestru stanu (status register). Po niewielkich zmianach w kodzie źródłowym biblioteki, wszystkie wyświetlacze jakie aktualnie posiadam już działają. Ponieważ jednak detekcja sterownika opiera się na nieudokumentowanych wartościach, w przyszłości mogą pojawić się moduły wyświetlaczy, które znowu nie będą działać.

W archiwum umieściłem przekompilowane (z poprawioną biblioteką) wsady programu sterującego w wersji 1.0/1.1 (kod źródłowy nie zmienił się) oraz programu do sprawdzania modelu przetwornika ADC. Ponadto dodałem katalog "ss\_oled-master", zawierający zmodyfikowany plik źródłowy biblioteki "ss\_oled".

## Spis elementów:

### REZYSTORY:

R1 - 75kΩ/1%  
R2 - 10kΩ/1%  
R3, R4 - 2.2kΩ

### KONDENSATORY:

C1 - 10uF/16V  
C2 - 22uF/10V  
C3 - 100nF/63V

### POTENCJOMETRY:

P1 - 20kΩ (Helitrim)

### DIODY:

D1 - 1N5817

### UKŁADY SCALONE:

U1 - LM1117-5.0

### PRZEŁĄCZNIKI:

S1-S3 - mikroprzełącznik

### MODUŁY:

MOD1 - Arduino Nano 3.0 (ATmega328/16 MHz)  
MOD2 - 16-bitowy przetwornik ADC ADS1115  
MOD3 - zegar RTC DS3231 z opcjonalną pamięcią EEPROM 24C32  
MOD4 - adapter ze złączem kart microSD  
MOD5 - 2 przekładniki 5V (JQC-3FF-S-Z)  
MOD6A, MOD6B - opcjonalny wyświetlacz OLED 128x64/128x32 (SSD1306)

### ZŁĄCZA:

CON1, CON5, J1 - goldpin 2x1 (męskie)  
CON2 - gniazdo zasilania DC-Jack  
CON3, CON4 - gniazdo śrubowe ARK 2-pin  
CON6 - goldpin 3x1 (żeńskie)  
MOD1 - goldpin 15x1+15x1 (żeńskie)  
MOD2 - goldpin 10x1 (żeńskie)  
MOD3, MOD4 - goldpin 6x1 (żeńskie)  
MOD5 - goldpin 3x1+4x1 (żeńskie), wysokość 8-9mm nad PCB  
oraz 4 druty miedziane o długości 17mm i średnicy 1.1-1.2mm.  
MOD6A, MOD6B - goldpin 4x1 (żeńskie)

### INNE:

J1 - zworka 2x1



