

## Ćwiczenie laboratoryjne 11

### Testy ogniw litowych przy użyciu mikrokontrolera Arduino

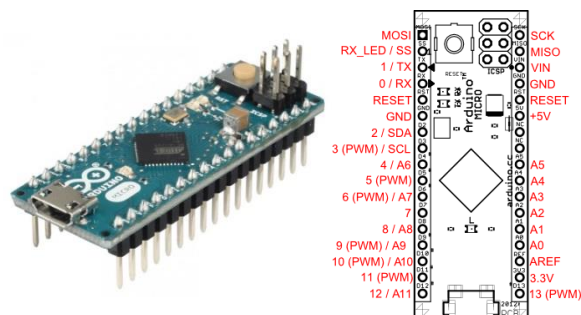
#### 1. Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z podstawowymi elementami elektronicznymi, dzięki którym można samodzielnie zaprojektować oraz skonstruować układ oparty na mikrokontrolerze Arduino, pozwalający na ładowanie komercyjnych akumulatorów litowych. W ćwiczeniu zastosowano komercyjnie dostępne ogniwo Samsung INR18650-25R.

#### 2. Część teoretyczna

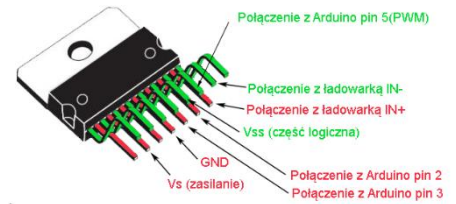
##### Elementy składowe układu

a) **mikrokontroler Arduino micro** – Główny element układu. Można porównać go do małego komputera, zdolnego to kontrolowania innych urządzeń. Aby płytką mogła działać, należy ją zaprogramować w dedykowanym oprogramowaniu (Arduino IDE) oraz wgrać na nią napisany program. Mikrokontroler Arduino wyposażony jest m.in. w następujące elementy:

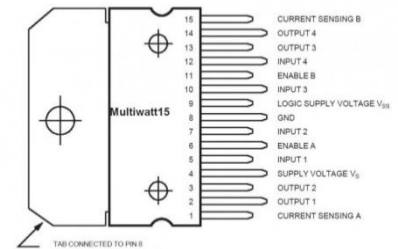


- złącze usb – wykorzystywane jest do zasilenia układu, zaprogramowania go oraz komunikacji z komputerem;
- piny – „nóżki” Arduino, do których podłączane są urządzenia sterowane przez układ.
  - piny cyfrowe (digital) – Ponumerowane od 2 – 12. Każdy z nich może być zaprogramowany, aby pełnił rolę wejścia, jak i wyjścia. Charakteryzują się tym, że jako piny wyjścia mogą mieć napięcie 0V lub 5V, natomiast jako wejścia akceptują również dwa poziomy napięć ok. 0V oraz napięcie między 2,5V a 5V. Sterują urządzeniami na zasadzie włącz/wyłącz.
  - piny analogowe (analog) – Ponumerowane od A0 do A5. Pełnią funkcję wejść. Mogą mierzyć napięcie od 0 do 5V. Mają rozdzielczość 10 bitów = rozpoznają 1024 poziomy napięcia, co oznacza dokładność ok. 0,005V.
  - piny PWM – część z pinów cyfrowych ma dodatkową funkcję – PWM (Pulse Width Modulation). Pozwala ona regulować moc jaka jest dostarczana do poszczególnych elementów, np. pozwala regulować prędkość sterowanego przez Arduino silnika.
- przycisk resetu (biały) – resetuje płytkę Arduino. Po jego przyciśnięciu wgrany program uruchomi się od nowa.
- diody LED – sygnalizator podłączonego napięcia do Arduino/wgrywania programu
- Mikrokontroler ATmega – „serce” płytki Arduino

**b) dwukanałowy sterownik silników L298N** – układ tranzystorowy, umożliwiający sterowanie silników i innych urządzeń. Sterownik posiada dwa kanały, co oznacza, że pozwala zasilać niezależnie dwa urządzenia – w rozważanym układzie wykorzystywany będzie tylko jeden kanał, do którego podłączona będzie ładowarka baterii. Sterownik posiada jeden pin zasilający ( $V_s$ ), jeden pin

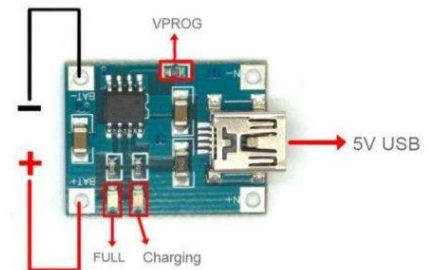


zasilający część logiczną ( $V_{ss}$ ), jeden pin masy (uziemia = GND), a także piny pozwalające na kontrolowanie wielkość napięcia wyjściowego, kierunku zasilania (przepływu prądu) i inne (ich znajomość nie jest wymagana w tym ćwiczeniu).

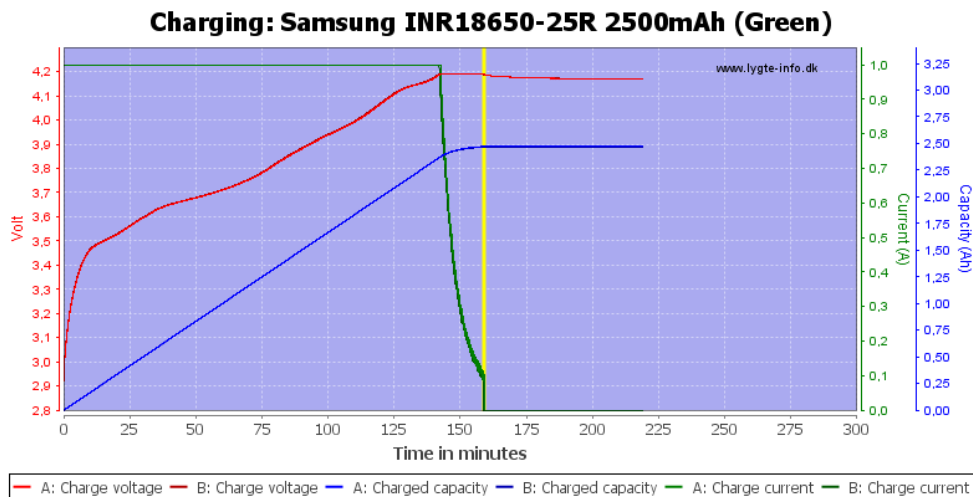


\* W układzie sterownik posiada radiator, który chroni przed nadmiernym przegrzaniem.

**c) ładowarka Li-ion TP4056** - moduł posiadający złącze mini USB (w układzie na zajęciach nieużywane), dwie diody LED – jedna sygnalizuje stan ładowania (czerwona), a druga informuje o naładowaniu baterii (niebieska/zielona). W przypadku, gdy złącze USB nie jest używane, zasilanie podłączone jest na wejścia IN+ oraz IN-. Napięcie wejściowe 4,5-5V. Ładowarka podaje na ogniwo stały prąd ok. 1A. Akumulator podłączony jest do wyprowadzeń z symbolami BAT+ oraz BAT-.



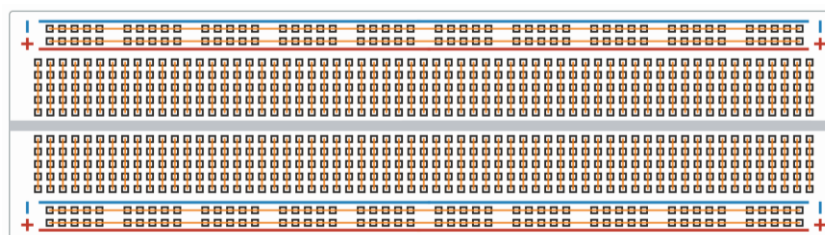
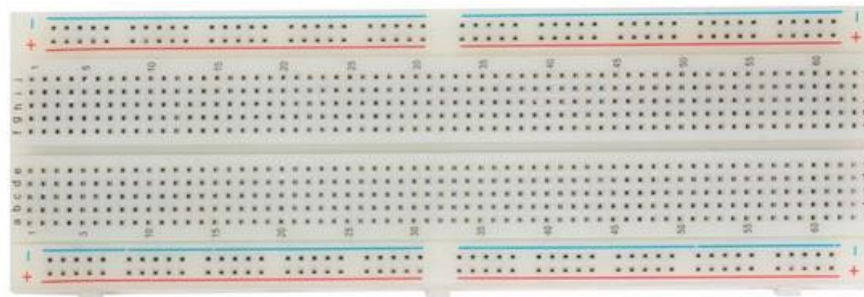
**d) ogniwo Samsung INR18650-25R** – komercyjne ogniwo litowe typu 18650 (średnica 18mm, długość 65mm) o pojemności 2500mAh. Skrót INR (naprzemiennie stosowany z NMC) oznacza zastosowanie tlenku  $\text{LiNi}_{1/3}\text{Co}_{1/3}\text{Mn}_{1/3}\text{O}_2$  jako materiał katodowy.



Rys. powyżej: profil ładowania ogniwa Samsung INR18650-25R.

Źródło: <https://tiny.pl/tqm8h>

- e) **plytka stykowa prototypowa** – płytka, na której umieszczone są wszystkie elementy układu. Płytkę pozwala na łatwy montaż/demontaż niezbędnych elementów, bez konieczności lutowania/rozlutowywania. Posiada 830 pól (rastrów o wymiarze 2,54mm), które połączone są ze sobą od kolektorami wg schematycznego rysunku poniżej. Ponadto płytka taka posiada 2 linie zasilające (+) oraz 2 linie uziemiające (-).



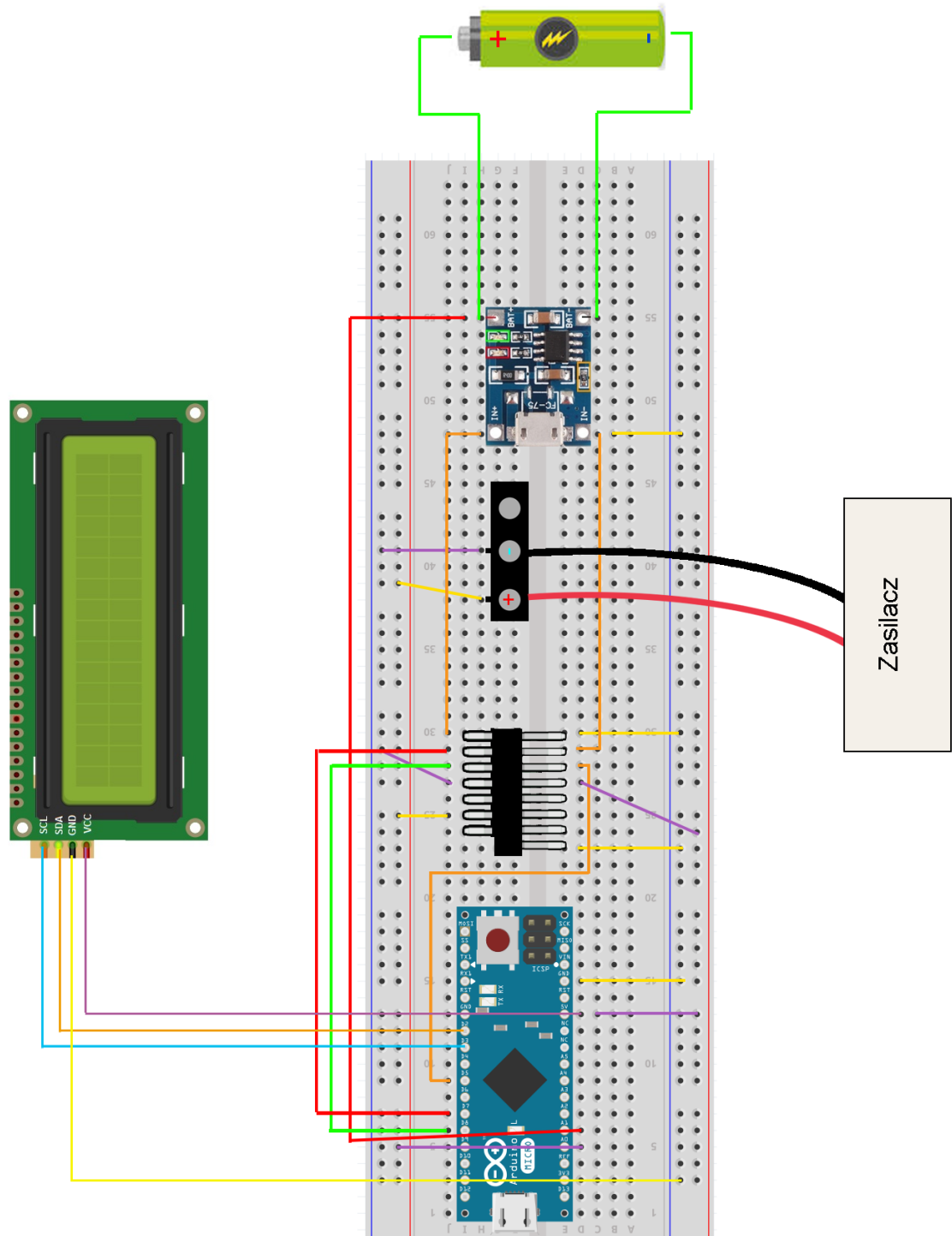
- f) **wyświetlacz LCD** – standardowy wyświetlacz o wymiarach 16x2 (16 kolumn, 2 rzędy), zgodny z mikrokontrolerami Arduino, zasilany napięciem 5V.



Posiada 4 podstawowe piny (zamieszczone na spodniej części wyświetlacza): GND (uziemienie), Vcc (pin zasilania 5V), SDA oraz SCL (piny, za pomocą których odbywa się odczyt danych z Arduino).

Poprzez odpowiednie zaprogramowanie (funkcja `lcd.print()`), można wyświetlić na nim dowolny komunikat.

### 3. Schemat układu ładowania ogniwa:



#### 4. Oprogramowanie ładowarki:

#### 5. Wykonanie ćwiczenia:

- a) Podłączyć elementy układu zgodnie ze schematem z punktu 3:
  - Połączyć mikrokontroler Arduino z ładowarką oraz sterownikiem:
    - Połączenie Arduino-sterownik: Pin5(Ard) – EnableB(sterownik)
    - Połączenie Arduino-ładowarka: PinA1(Ard) – Bat+(ładowarka)
  - Podłączyć ładowarkę do zasilania
    - Połączenie ładowarka(+)-sterownik: In+(ładowarka) – Output4(sterownik)
    - Połączenie ładowarka(-)-sterownik: In-(ładowarka) – Output3(sterownik)
  - Podłączyć ogniwo do ładowarki
    - Połączenie ładowarka-ogniwo: Bat+(ładowarka) – katoda ogniwa
    - Połączenie ładowarka-ogniwo: Bat-(ładowarka) – anoda ogniwa
- b) Za pomocą multimetru odczytać napięcie ogniwa Samsung 25R.
- c) Podłączyć układ do komputera za pomocą kabla USB **(WAŻNE! Do podłączenia kabla USB użyć port na Arduino, nie na ładowarce!)**
- d) Uruchomić program `Ladowarka_Li_ion.ino` znajdujący się w folderze *Akumulatory* >> *Ladowarka\_Li\_ion* na pulpicie. Wejść w zakładkę `settings.h`. Ustawić górny i dolny zakres ładowania oraz częstość czytania punktów.

```

Ladowarka_Li_ion - settings.h | Arduino 1.8.5
Plik Edytuj Szkic Narzędzia Pomoc

Ladowarka_Li_ion charger.h settings.h

#ifndef SETTINGS_H
#define SETTINGS_H

namespace Settings
{
    // ---- limit napięcia baterii [V] ---- //
    constexpr float voltageUpperLimit = 4.0;
    constexpr float voltageLowerLimit = 2.5;

    // ---- częstość zapisu [1/min] ---- //
    constexpr int freq = 30;

    // ---- wejścia analogowe odczytu napięcia ---- //
    constexpr int aInBattery = 1;
    constexpr int aInPower = 0;

    // ---- sterowanie LM298N ---- //
    constexpr int dOutIN1 = 2;
    constexpr int dOutIN2 = 3;
    constexpr int dOutPWM = 5;
    constexpr int PWMValue = 255;

    // --- sterowanie LCD --- //
    constexpr int dOutLCDBacklight = 7;
}

#endif // SETTINGS_H
    
```

- e) Skompilować i wgrać program na płytkę Arduino. Służą do tego dwa przyciski w lewym górnym rogu programu: . W przypadku błędu, sprawdzić, czy płytka została poprawnie odczytana przez program (*Narzędzia* >> *Płytki* : tutaj powinna być widoczna nazwa Arduino Micro).
- f) Uruchomić program `LiIonCharger.exe`, znajdujący się w folderze *Akumulatory* >> *Gui* na pulpicie. Połączyć się z kontrolerem i poczekać na potwierdzenie.
- g) Utworzyć plik tekstowy, do którego mają być zapisywane dane, zbierane podczas ładowania ogniwa.

- h) Podpiąć zasilanie. **Upewnić się, że kable od zasilacza zostały prawidłowo podłączone. Czerwony kabel do przyłącza +, czarny do -.**
- i) Ustawić napięcie na zasilaczu na ok. 6,8V. **W przypadku, gdy na ładowarce świecą się obie diody (czerwona i niebieska) natychmiast odłączyć zasilanie!**

## 6. Czego nie należy robić (!)

- Odłączać lub przyłączać elementów elektronicznych przy podpiętym zasilaniu (zasilacz lub ładowarka 12V).
- Odłączać lub przyłączać elementów elektronicznych przy podpiętym zasilaniu Arduino (kabel USB).
- Podłączać zasilania innego niż wskazane przez prowadzącego.
- Wypinać elementów, które nie zostały wskazane w instrukcji lub przez prowadzącego (w szczególności dotyczy to sterownika, płytki Arduino oraz ładowarki).
- Dołączać do układu dodatkowych elementów.

## 1. Przygotowanie sprawozdania

- a) Wyznaczyć krzywą ładowania ogniwa Samsung 25R jako funkcję napięcia od czasu  $V(t)$ .
- b) Czy krzywa ładowania ogniwa odpowiada krzywej podawanej w specyfikacji komercyjnej? Jeżeli nie, z czego to może wynikać?

## Literatura

1. W. Andrews *Zrób to sam w Arduino*, wyd. PWN, Warszawa 2018
2. S. Monk *Arduino dla początkujących. Podstawy i szkice*, wyd. Helion 2018
3. <https://forbot.pl/blog/kurs-arduino-srodowisko-jak-zaczac-programowac-id936>  
[dostęp: 17.12.2018]