Podstawy Internetu Rzeczy
Instrukcja do zajęć laboratoryjnych

Laboratorium 1

Wprowadzenie do laboratorium. Szkolenie BHP
Wprowadzenie do pracy z zestawami Arduino

Opracował opiekun przedmiotu dr inż. Krzysztof Chudzik

Wydział Informatyki i Telekomunikacji
Politechnika Wrocławska

Wrocław, 2025.09.30 18:49:42

Spis treści

Lista zadań

1 Wprowadzenie do laboratorium. Szkolenie BHP

Proszę, aby Prowadzący podał do wiadomości Studentów:

- Zasady bezpiecznej pracy w laboratorium, w tym:
  - regulamin korzystania z laboratorium,
  - podstawowe zasady obowiązujące w laboratorium,
  - lokalizację głównego wyłącznika energetycznego oraz gaśnic,
  - wskazał drogę ewakuacji z laboratorium w przypadku zagrożenia.
- Na podstawie karty przedmiotu:
  - wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych,
  - cele przedmiotu,
  - przedmiotowe efekty uczenia się,
  - treści programowe laboratorium (treści wykładu zostaną zaprezentowane na wykładzie),
  - zasady oceny osiągnięcia przedmiotowych efektów uczenia się, czyli **termin i warunki zaliczenia laboratorium**.
- Termin konsultacji i sposób kontaktu z Prowadzącym zajęcia.

Pewne wybrane zagadnienia zostaną najpierw wprowadzone na laboratorium, a następnie wiedza na ich temat zostanie poszerzona w ramach wykładu. Wiedza wymagana do wykonania ćwiczeń laboratoryjnych zostanie podana w materiałach do laboratorium, przez Prowadzącego laboratoria oraz przez wskazanie w materiałach lub przez Prowadzącego dodatkowych materiałów, z którymi należy zapoznać się przed laboratorium. Na wykładzie będziemy odwoływać się do praktycznych umiejętności i doświadczeń zdobytych przez Państwa na laboratorium, co ułatwi zrozumienie wykładanego materiału. Jest to działanie zamierzone i zaplanowane przez Opiekuna kursu prowadzącego wykład.

Student, w ramach przygotowania do zajęć, ma obowiązek zapoznać się z instrukcją do laboratorium i materiałami w niej wskazanymi przed każdym laboratorium, aby efektywnie wykorzystać sprzęt laboratoryjny i nie spowodować jego uszkodzeń. Jeśli Prowadzący laboratorium stwierdzi, że Student nie jest przygotowany do zajęć lub narusza zasady pracy w laboratorium i posługiwania się sprzętem laboratoryjnym, ma obowiązek odsunąć Studenta od zajęć.

2 Prezentacja zestawów laboratoryjnych

Pierwsza część laboratorium kursu Podstawy Internetu Rzeczy odbywa się z wykorzystaniem zestawu laboratoryjnego wyposażonego w płytkę Arduino i dodatkowe elementy pozwalające poznawać wybrane zagadnienia interakcji urządzeń, w tym urządzeń Internetu Rzeczy, z otoczeniem, w szczególności z człowiekiem. Dalej ten zestaw laboratoryjny będziemy nazywać w skrócie „zestawem Arduino", lub po prostu „zestawem".

Tę część laboratorium rozpocznie zatem prezentacja zestawu Arduino. Proszę, aby prezentacji dokonał Prowadzący laboratorium z wykorzystaniem slajdów: „Zestawy laboratoryjne z Arduino - Wprowadzenie". Slajdy te są udostępnione w ramach platformy *ePortal* i Studenci uczestniczący w laboratorium powinni z nich korzystać, ponieważ są podstawowym dokumentem opisującym konfigurację zestawu.

3 Omówienie zasad obchodzenia się z zestawami laboratoryjnymi Arduino

Każde kolejne laboratorium przebiegać będzie z zachowaniem zasad omówionych poniżej.

3.1 Rozpoczynanie pracy z zestawem

Aby chronić stanowiska w laboratorium i same zestawy przed dewastacją, Studenci przez cały semestr zajmują te same numerowane stanowiska. Prowadzący prowadzi ewidencję tych miejsc. Zmiana ich bez zgody Prowadzącego jest zabroniona.

Po rozpoczęciu zajęć Prowadzący rozdaje Studentom zestawy. Każdy Student obecny w laboratorium otrzymuje jeden zestaw. Zestawy są numerowane, a obowiązkiem Prowadzącego, jest prowadzenie ewidencji, który zestaw został przekazany danemu Studentowi. Aby uprościć procedury, przyjmuje się zasadę, że Student otrzymuje zestaw z numerem swojego stanowiska. W przypadku awarii tego zestawu, Prowadzący wręcza Studentowi inny zestaw, odnotowując jego numer.

Student w czasie zajęć odpowiada za swoje stanowisko i zestaw, który otrzymał na czas zajęć. Zestawy są wrażliwym sprzętem elektronicznym i jako takie powinny być użytkowane delikatnie. Zabrania się jakiegokolwiek manipulowania przy zestawach, w szczególności zmiany konfiguracji połączeń układów elektronicznych, używania nadmiernej siły lub gwałtownych ruchów (naciskania, lub obracania) wobec przycisków, pokręteł, itp.

Szczególnie wrażliwa ma uszkodzenia jest sonda temperatury z tyłu obudowy. Zabrania się zginania oraz ściskania części obudowy sondy z nadmierną siłą prowadzącą do odkształcenia obudowy lub uszkodzenia

mocowania w obudowie. Jeśli zachodzi potrzeba podgrzania sondy dłonią, to ściskamy palcami delikatnie wyłącznie odsłoniętą część metalową sondy.

Student, ma obowiązek sprawdzić na początku zajęć, czy wszelkie elementy wyposażenia stanowiska znajdują się we właściwym stanie i czy zestaw, który otrzymał jest sprawny. Jeśli tak nie jest, informuje o tym Prowadzącego, który podejmuje decyzję stosownie do okoliczności.

Zestaw powinien mieć wgrany wcześniej program testujący, umożliwiający ocenę stanu technicznego urządzenia. Jeśli tak nie jest, należy o tym poinformować Prowadzącego, który odnotuje ten fakt.

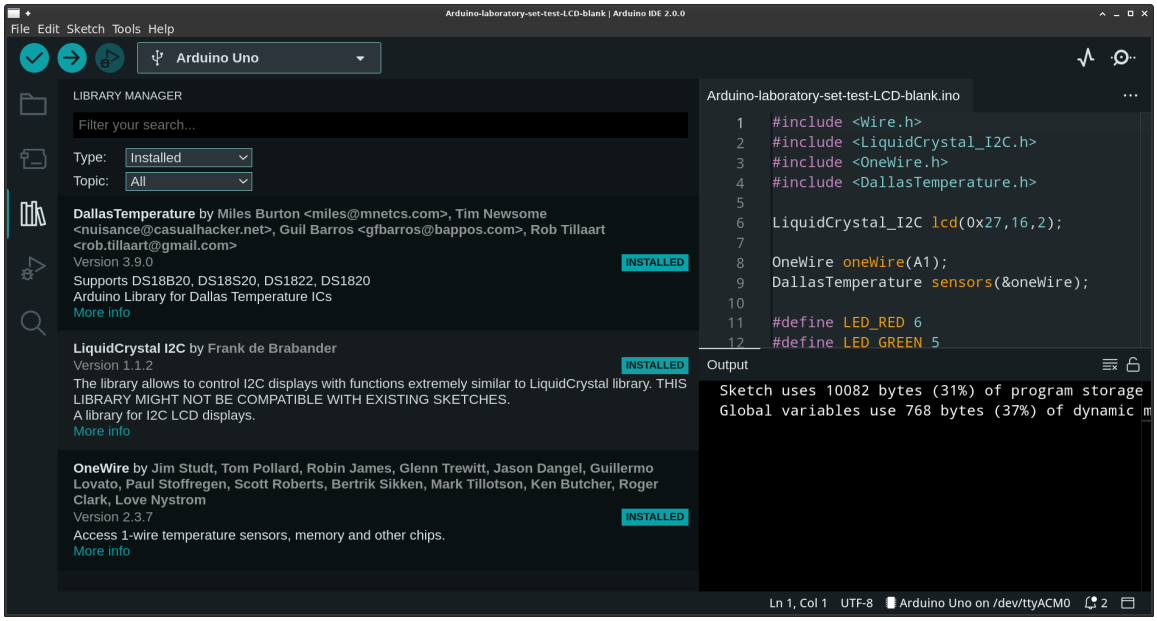
3.2 Program testujący zestaw

Kod źródłowy programu testującego zestaw dostępny jest na platformie *ePortal*. Program należy wgrać i uruchomić na zestawie laboratoryjnym.

Program przygotowany został w dwóch wersjach:

- `Arduino-laboratory-set-test` - podstawowa wersja programu,
- `Arduino-laboratory-set-test-LCD-blank` - wersja, która po zakończeniu działania wygasza ekran LCD.

Do uruchomienia programu wymagana jest instalacja w środowisku Arduino IDE bibliotek wykazanych na Rysunku 1. Opis zarządzania bibliotekami zawarty jest w instrukcji Installing libraries (link).



Rysunek 1: Biblioteki wymagane do uruchomienia programu testowego.

Proszę zapoznać się z nagraniem „Procedura testowa zestawu Arduino" dostępnym na platformie *ePortal*. Wszelkie odstępstwa od zachowania programu, które przedstawiono na nagraniu, należy traktować jako uszkodzenie zestawu i zgłosić Prowadzącemu zajęcia laboratoryjne.

3.3 Zgłaszanie usterek zestawów

Student ma obowiązek zgłaszania wszelkich usterek zestawu i usterek na stanowisku laboratoryjnym. Jak wspomniano, przed rozpoczęciem zajęć Student ma obowiązek przetestować zestaw i zweryfikować czy wszelkie elementy wyposażenia stanowiska znajdują się we właściwym stanie.

Uszkodzenia stwierdzone na stanowisku lub uszkodzenia zestawu, nie zgłoszone zaraz po rozpoczęciu zajęć, uznaje się za dokonane przez Studenta, który zajmował stanowisko. Sytuacje wątpliwe rozstrzyga Prowadzący, stosownie do zaistniałych okoliczności.

Należy jednak pamiętać, że zestaw, jak każde urządzenie, może ulec awarii podczas normalnej eksploatacji. Jeśli doszło do takiej sytuacji po rozpoczęciu zajęć, Student ma obowiązek zgłosić taki fakt Prowadzącemu. Świadome zatajenie takiej sytuacji, będzie traktowane jak świadome uszkodzenie zestawu. Prowadzący, po stwierdzeniu, że zaistniała sytuacja jest przypadkiem losowym, wydaje Studentowi inny, sprawny zestaw.

W przypadku stwierdzenia, że Student świadomie, lub na skutek braku właściwego przygotowania się do zajęć przez Studenta, poczynił szkody na stanowisku lub uszkodził zestaw, Prowadzący odsuwa Studenta od zajęć i informację o zaistniałej sytuacji przekazuje do stosownych władz.

3.4 Elementarne zasady posługiwania się zestawem

- Jeśli przygotowany program funkcjonuje na tyle źle, że konieczny jest reset Arduino, resetujemy urządzenie naciskając czarny przycisk.
- Nie wolno w tym celu odłączać kabla USB.
- Należy liczyć się z tym, że wyświetlacz LCD 16x2 nie zmieni wyświetlanej zwartości ekranu i będzie pokazywał to co przed resetem.
- Program, po poprawieniu i ponownym uruchomieniu, powinien sam czyścić ekran wyświetlacza LCD. Oznacza to, że nie powinniśmy zakładać, że ekran będzie wyczyszczony samoczynnie po starcie programu.

3.5 Zdawanie zestawów na zakończenie zajęć

Przed zakończeniem zajęć, obowiązkiem Studenta jest wgranie do zestawu programu testującego zestaw i przetestowanie działania zestawu. Po sprawdzeniu, oddaje zestaw, z wgranym programem testującym, Prowadzącemu i informuje Prowadzącego o stanie technicznym zestawu. Prowadzący odnotowuje wszelkie uwagi odnoszące się do stanu technicznego zestawu. Prowadzący odpowiada za informacje o aktualnym stanie technicznym zestawów i przekazuje informacje o ewentualnych uszkodzeniach Opiekunowi przedmiotu.

4 Programowanie Arduino z wykorzystaniem IDE

Arduino IDE powinno być wcześniej zainstalowane na komputerowych stanowiskach laboratoryjnych. Studenci mogą też zainstalować oprogramowanie na swoich komputerach.

Oprogramowanie Arduino IDE należy pobrać ze strony Arduino Downloads (link). Do zajęć proszę pobrać najnowszą wersję Arduino IDE 2 stosownie do posiadanej wersji systemu operacyjnego.

Strona Arduino IDE (link) zawiera zestaw bardzo szczegółowych instrukcji (ang. tutorials) wyjaśniających jak należy posługiwać się Arduino IDE. Proszę zapoznać się z tymi instrukcjami.

5 Podstawowa struktura programu w Arduino IDE

Programowanie Arduino odbywa się, na ogół, z wykorzystaniem języka C lub C++. W ramach wykładu zostanie omówiona dokładnie struktura programu i podstawowe techniki programistyczne. Na potrzeby laboratorium podane zostaną jedynie informacje niezbędne do rozpoczęcia programowania zestawów Arduino.

Podstawowa struktura programu mikrokontrolera przedstawiona została jako Kod 1.

Kod 1: Struktura programu mikrokontrolera.

```
1  //DO NOT COMPILE - FOR INFORMATION ONLY!
2  #include <your_header1.h> //Wymagane pliki nagłówkowe
3  #include <your_header2.h>
4
5  int main(void)
6  {
7      //Kod inicjalizacyjny.
8
9      while (true) //Nieskończona pętla
10     {
11
12         //Główny kod programu wykonywany cyklicznie.
13     }
14 }
```

Program składa się więc z dwóch zasadniczych części.

Pierwszą jest kod inicjalizacji. Wykonywany jest on jeden raz po starcie programu. Jego zadaniem jest skonfigurowanie nie tylko zmiennych czy obiektów w programie, ale, przede wszystkim, skonfigurowanie trybów pracy i stanów układów wewnętrznych i portów mikrokontrolera.

Drugą częścią programu jest główny kod programu wykonywany cyklicznie. Zwrócić należy uwagę, że, w ogólnym założeniu, wykonywanie kodu mikrokontrolera nie kończy się nigdy, o ile urządzenie z mikrokontrolerem jest zasilane i nie wystąpi reset mikrokontrolera (sprzętowy lub programowy). Stąd główny kod programu umieszczony jest wewnątrz niekończącej się pętli `while`.

Powyższa struktura ma swoje odbicie kodzie szablonowym, który prezentuje środowisko programistyczne Arduino IDE, przedstawionym jako Kod 2.

Kod 2: Szablon kodu dla Arduino prezentowany w Arduino IDE.

```
1  void setup() {
2    // put your setup code here, to run once:
```

```
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Nie ma tutaj funkcji `main()`, gdyż w tym środowisku programistycznym została ona ukryta przed użytkownikiem środowiska. Wywołuje ona jednak dwie funkcje `setup()` i `loop()`. Funkcja `setup()` wykonywana jest jednokrotnie po starcie programu i w wewnątrz niej umieszczony powinien być kod inicjalizacyjny. Funkcja `loop()` wywoływana jest cyklicznie i w jej wnętrzu powinien znaleźć się główny kod programu.

Ponadto, poza ciałami funkcji `setup()` i `loop()`, zgodnie z zasadami jezków C i C++, możemy umieścić dyrektywy preprocesora (w szczególności `#include`), deklaracje i definicje innych funkcji oraz deklaracje zmiennych i obiektów globalnych, itd.

Aby lepiej poznać strukturę kodu Arduino można skorzystać ze środowiska VS Code. Po utworzeniu projektu dla Arduino, możemy wyszukać odniesienia (ang. *References*) do funkcji `loop()` i `setup()`. Ich deklaracje możemy znaleźć w plku `Arduino.h`, natomiast użycie w pliku `main.cpp`, we wnętrzu funkcji `main()`. Oba pliki znajdują się w folderach będących częścią Arduino IDE. Fragment pliku `main.cpp` zawierający kod funkcji `main()` z wywołaniami funkcji `setup()` i `loop()` zawiera Kod 3.

Kod 3: Fragment pliku `main.cpp` zawierający definicję funkcji `main()`.

```
1 //DO NOT COMPILE - FOR INFORMATION ONLY!
2 int main(void)
3 {
4     init();
5
6     initVariant();
7
8 #if defined(USBCON)
9     USBDevice.attach();
10 #endif
11
12     setup();
13
14     for (;;)
15     {
16         loop();
17         if (serialEventRun)
18             serialEventRun();
19     }
20
21     return 0;
22 }
```

W ten sposób buduje się swoisty język programowania Arduino. Jego dokumentację można należć na stronie Language Reference (link).

Mając tą wiedzę, wiemy, że możemy pisać program w języku C lub C++.

Przed rozpoczęciem pisania własnego kodu, spójrzmy na bardzo prosty przykład. Kod 4 zawiera przykładowy program migający diodą świecącą (LED) na płytce Arduino. Program dostępny jest z Menu: File > Examples > 01.Basics > Blink.

Kod 4: Przykład z Arduino IDE: Blink.ino

```
1 // the setup function runs once when you press reset or power the board
2 void setup() {
3   // initialize digital pin LED_BUILTIN as an output.
4   pinMode(LED_BUILTIN, OUTPUT);
5 }
6
7 // the loop function runs over and over again forever
8 void loop() {
9   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
10  delay(1000); // wait for a second
11  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
12  delay(1000); // wait for a second
13 }
```

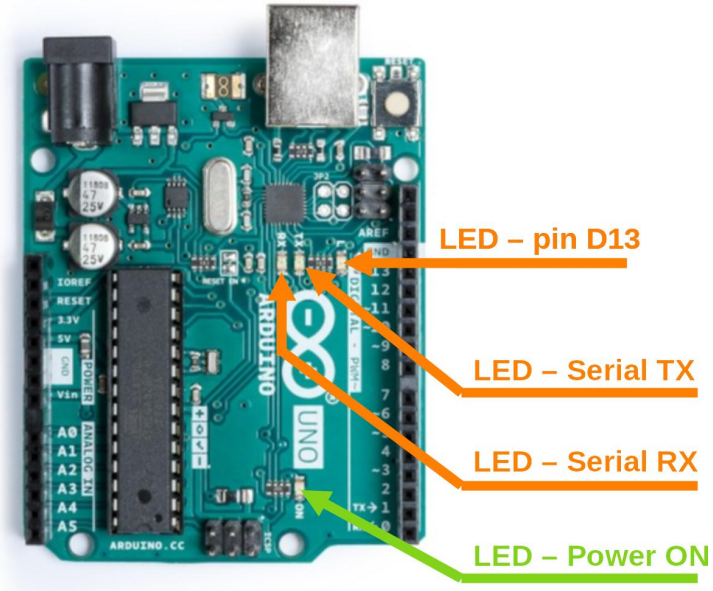
Najpierw mała uwaga językowa. Termin angielski „pin”, przyjęty też w literaturze polskiej, może być rozumiany jako:

- wyprowadzenie („nóżka”) mikrokontrolera,
- pojedyncze wyprowadzenie w złączu na płytce Arduino, do którego możemy przyłączyć przewód łączący z zewnętrznym elementem.

W dokumentacji lub artykułach w języku polskim niekiedy używa się pojęcie „styk” tłumacząc „pin”.

W przypadku płytki Arduino UNO, odpowiednie wyprowadzenia mikrokontrolera ATmega328P są bezpośrednio podłączone do odpowiednich wyprowadzeń złączy na brzegach płytki Arduino. Zajrzyj ponownie do prezentacji „Zestawy laboratoryjne z Arduino - Wprowadzenie", aby to zweryfikować. Poszukaj też w Internecie schematu elektrycznego Arduino Uno i spróbuj samodzielnie sprawdzić. Wobec powyższego, konfiguracja pinu mikrokontrolera i pinu płytki Arduino jest tym samym.

Na rysunku 2 przedstawiono lokalizację wbudowanych diod świecących na płytce Arduino. Kolory strzałek i napisów odpowiadają kolorom diod na płytce Arduino wykorzystywanej w zestawie laboratoryjnym. Dioda, którą steruje program Blink, to dioda wbudowana w płytkę „LED - pin 13".



Rysunek 2: Diody świecące (LED) na płytce Arduino.

Funkcja `setup()` konfiguruje pin sterujący diodą świecącą na płytce jako wyjście. Funkcja `pinMode()` konfiguruje określony pin, aby zachowywał się jak wejście lub wyjście. Funkcja przyjmuje dwa parametry: numer pinu i tryb. Numer pinu podany jest w postaci identyfikatora `LED_BUILTIN`, który wprowadzony jest dyrektywą preprocesora `#define`, a przyjmuje wartość 13, czyli numer trzynastego pinu cyfrowego (D13). Podobnie, identyfikatorem też wskazano, że pin będzie pracował jako wyjście. Do tego pinu, w odpowiedni sposób, podłączona jest dioda LED i rezystor zabezpieczający ją.

Główny kod programu wykonywany cyklicznie, zdefiniowany w funkcji `loop` wykorzystuje dwie funkcje. Funkcja `digitalWrite()` ustawia na pinie podanym jako pierwszy argument stan logiczny (i tym samym napięciowy) wysoki (identyfikator `HIGH`) lub niski (identyfikator `LOW`), który podaje się jako drugi argument funkcji. Stan wysoki powoduje świecenie diody, stan niski, że dioda jest zgaszona. Funkcja `delay()` wstrzymuje program na czas (w milisekundach) podany jako parametr. Pamiętajmy, że 1 sekunda to 1000 milisekund. Teraz już łatwo zinterpretować działanie funkcji `loop()`. Zapala diodę na 1 sekundę, a następnie gasi na 1 sekundę. Ponieważ wywoływana jest cyklicznie, to otrzymujemy efekt ciągłego migania diody.

Zadanie 1: Sterowanie diodą świecącą na płytce Arduino

Zmień program Blink.ino (Kod 4) tak, aby dioda świecąca realizowała sekwencję zadaną przez Prowadzącego.

Zadanie 2: Wgranie programu testującego i przetestowanie zestawu

Na zakończenie zajęć należy wgrać program testujący zestaw i zweryfikować nim poprawność funkcjonowania zestawu. O wszelkich nieprawidłowościach proszę poinformować Prowadzącego zajęcia.
Uwaga: To zadanie należy wykonywać na każdych kolejnych zajęciach z zestawami Arduino na ich zakończenie.

6 Czy chcesz wiedzieć więcej o Arduino?

Nasz kurs nie jest kursem o Arduino samym w sobie. Arduino jest tylko narzędziem, pozwalającym wejść w świat mikrokontrolerów i podstawowych urządzeń peryferyjnych, które są centralnym elementem dużej części urządzeń Internetu Rzeczy, szczególnie tych niewielkich.

Gdyby ktoś chciał pogłębić swoja wiedzę na temat Arduino, to polecam materiały zamieszczone na witrynie Forbot (link):

- Seria artykułów: Arduino, co w środku piszczy (link).

- Kursy zawarte na stronie: Kursy elektroniki i programowania (link). Są tam, między innymi, różne poziomy kursu Arduino oraz, dla osób bardziej zainteresowanych sprzętem, kursy elektroniki.