

Przygotowanie do Dart-Plus

Jakub Delicat

8 marca 2022

1 Wprowadzenie

Zadanie ma na celu sprawdzić waszą zdolność szybkiego uczenia się, znajomości języka C++ i elektroniki.

2 Zagadnienia do nauki

Zagadnienia potrzebne do zrobienia zadania testowego:

- jak podłączyć i zaświecić LEDem RGB.
- jak podłączyć i odczytać przycisk. Poradnik do wejść i wyjść.
- jak działa maszyna stanów np. tutaj.

3 Treść zadania 1

Stwórz układ, który będzie pracował przy użyciu maszyny stanów. Podłącz 2 przyciski do Arduino do pinów cyfrowych 2 i 4 oraz LED RGB do pinów 4 - red, 5 - green, 6 - blue. Napisz program, w którym za pomocą jednego przycisku będziesz zmieniał kolor diody, a za pomocą drugiego będziesz wyłączał diodę. Dodatkowo spróbuj zrobić na ostatnim stanie dyskotekę z diody RGB.

3.1 Wymagania

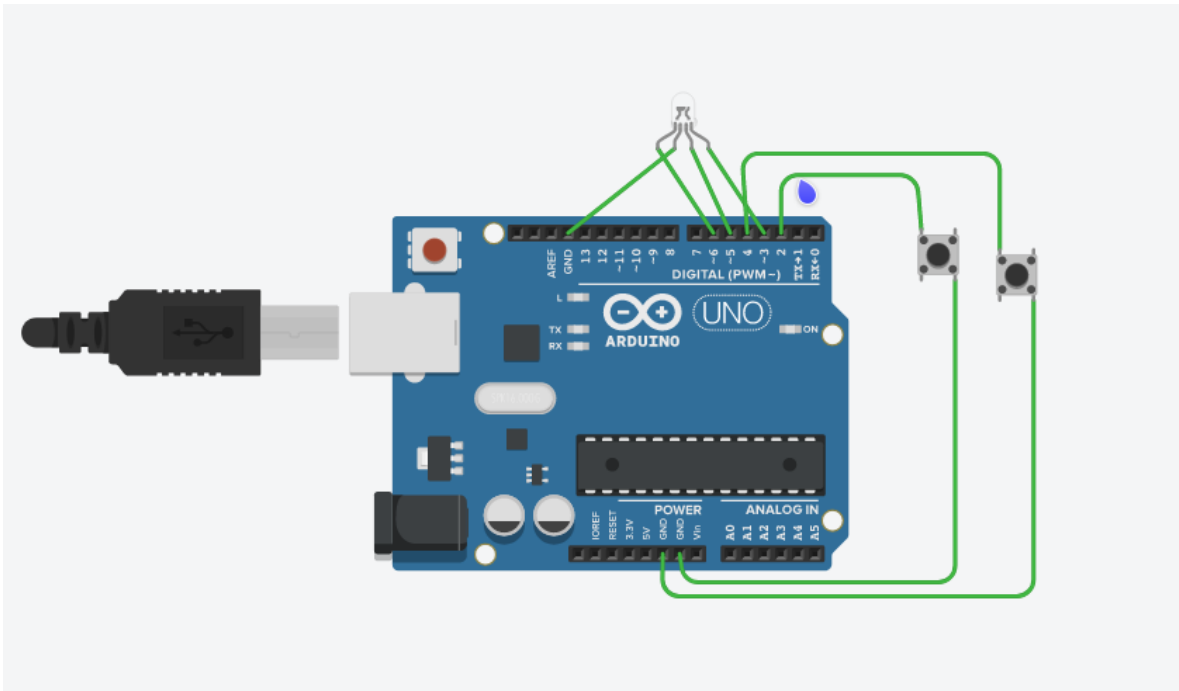
- Stany zapisz do typu enum:

```
enum States {UNKNOWN, DISABLED, LEDR, LEDG, LEDB, DISCO};
```

- Do sprawdzania stanu użyj switch:

```
enum States {    UNKNOWN,    // nie wiadomo co sie dzieje
                DISABLED,    // wyłączona dioda
                LEDR,        // zapal czerwona
                LEDG,        // zapal zielona
                LEDB,        // zapal niebieska
                DISCO};      // zrob dyskoteke

States state = UNKNOWN;
// program...
switch(state){
    case UNKNOWN:
        ...
        break;
    case DISABLED:
        ...
        break;
}
```



Rysunek 1: Sposób podłączenia schematu

4 Treść zadania 2

Nauczmy się współpracować w takim razie. Używać będziemy Githuba. Na tym repozytorium stwórzcie własną gałąź i przejdź na nią. Tam wrzucicie swój kod w pliku *main.cpp*. Przypomnienie:

- Stwórz swój kod ssh i dołącz go do Githuba. Poradnik od Githuba.
- Aby ściągnąć repozytorium użyj komendy **git clone <url>**. Koniecznie użyj protokołu **ssh**, nie **https**.
- Aby stworzyć gałąź użyj **git checkout -b <nazwa>** -b tworzy nową gałąź
- Aby dodać pliki do commita używamy **git add <nazwa-pliku>**
- Aby potwierdzić zmiany i zapisać commita z wiadomością co zmieniliśmy używamy **git commit -m <wiadomosc>**
- Aby wrzucić swojego commita na serwer Githuba używamy **git push**. Zapewne git poprosi was o flagę *-up-stream* co oznacza, że wrzuca nową gałąź do repozytorium **git push -u origin <nazwa-brancha>**

4.1 Wymagania

- Twoja gałąź powinna nazywać się zgodnie ze schematem: **feature/deli-state-machine** (feature/<twój nick>-<co robisz na tej galezi>)
- Twój plik *main.cpp* powinien znajdować się w folderze z twoim imieniem i nazwiskiem: *jakub_delicat/main.cpp*

5 Treść zadania 3

Popracujemy na repo. W zakładce issues jest problem, który musisz rozwiązać. Stwórz commita o odpowiedniej nazwie, która połączy się z problemem na Gitcie.

5.1 wymagania

Treść commita powinna z rozwiązaniem problemu wyglądać w następujący sposób:

- #<nr_issue> <name_issue>
 - changes
 - changes

Na przykład:

- #69 Add test to esp communication
 - initialize test/communication_test.cpp
 - include test library into requirements in CMakeLists.txt

6 Treść zadania 4

To co kotki lubią najbardziej. Zaimplementujemy twój kod do projektu. Na gitcie musisz stworzyć pull requesta do mastera. Na Githubie wchodzisz w zakładkę pull requests, klikasz przycisk new pull request. Wybierasz żeby twoja gałąź połączyła się z masterem. Potwierdzasz i czekasz na odpowiedź. Warunkiem zaliczenia PTMów jest to żeby każdy wiedział co się dzieje w każdym miejscu w projekcie dlatego ważne jest code review i czytanie czyjegoś kodu.

6.1 Wymagania

Każdy musi przeczytać wszystkie pull requesty od innych osób. Ma to za zadanie zaznajomić się z kogoś kodem jak i rozwinąć nasze umiejętności zwracając uwagę na błędy czy nieprawidłowe praktyki. Jeśli czegoś nie rozumiecie zapytajcie dodając komentarz do danej linijki w kodzie. Jeśli uważacie, że ktoś ma błąd napiszcie to w komentarzu. Jeśli uważacie, że ktoś mógł coś lepiej zrobić zasugerujcie mu to w komentarzu do danej linijki kodu.

Powodzenia w developowaniu!