

Blynk – sterowanie przez Internet w 5 minut

Budowanie oraz programowanie urządzeń elektronicznych staje się coraz łatwiejsze ze względu na dostępność opensourcowych bibliotek programistycznych oraz gotowych platform z procesorem, np. Arduino, Raspberry PI i innych. Wykonanie projektu z wyświetlaczem może się ograniczyć do pobrania odpowiedniej biblioteki, nalożeniu shielta i już jest gotowe coś, co kiedyś zajęłoby bardzo dużo czasu. Nasuwa się pytanie, czy podobnie stanie się z urządzeniami IoT? Poprawna odpowiedź to – już tak się stało!

Mamy dostępnych wiele bibliotek i platform do sterowania urządzeniami przez Internet, pobierania z nich danych (np. wyników pomiarów), archiwizacji. Wystarczy tylko je umiejętnie poskładać jak popularne klocki Lego. Jest to o tyle interesujące rozwiązanie, że gotowe platformy zastępują pracę ludzi z różnych dziedzin. Dla przykładu, gdybyśmy chcieli wykonać projekt termometru z dostępem przez telefon i stronę www z możliwością zapisywania pomiarów do bazy danych i wyświetlaniem wykresów, to potrzebujemy mieć kompetencję z wielu różnych dziedzin lub kilku fachowców do pomocy.

Przedstawię przykładową platformę ułatwiającą tworzenie projektów IoT, którą jest Blynk. Jej sloganem reklamowym jest, że „można zamigać LED przez Internet w mniej niż 5 minut” i moim zdaniem to jest prawda.

Jak działa Blynk?

Zasada działania jest prosta – instalujemy specjalną aplikację na swoim smartfonie lub tablecie. Tworzymy tam projekt i wybieramy odpowiednią platformę uruchomieniową. Wgrywamy na nią program pobrany ze strony Blynka i po uruchomieniu nasz sprzęt może już być sterowany z telefonu. Oczywiście, jest to najprostszyp przypadk. Możemy modyfikować przykładowe programy i dodawać tam, cokolwiek chcemy – obsługę sensorów temperatury, wyświetlacze itp. Pełną listę wspieranych platform możemy znaleźć na stronie <http://docs.blynk.cc/#supported-hardware>. Co ważne, wśród wymienionych platform są płytki Arduino oraz

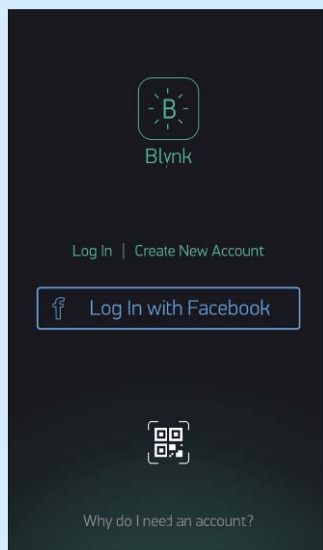
różne rozwiązania z procesorem ESP8266, który ostatnimi czasy zyskuje coraz większą popularność.

Moim zamierzeniem jest, aby opis doprowadził do uzyskania użytecznego urządzenia, więc do omówienia tworzenia projektu wykorzystamy praktyczny opis urządzenia ESP RELAY, który będzie opublikowany w kolejnym wydaniu „Elektroniki Praktycznej”. Zawiera on moduł ESP07 z mikrokontrolerem ESP8266. Musimy mieć poprawnie skonfigurowane środowisko Arduino IDE do jego obsługi (opis znajdziemy w artykule na jego temat).

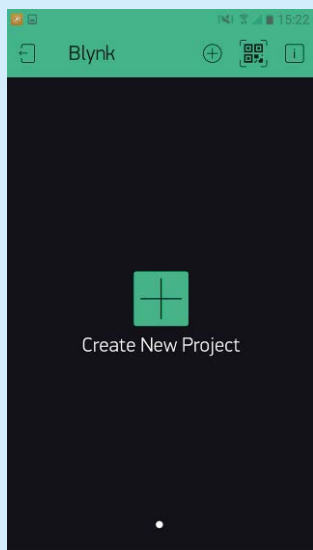
Rozpoczynamy

Pobieramy Blynka z GooglePlay lub iOS Store i uruchamiamy. Przywita nas okno jak na **rysunku 1**. Zakładamy nowe konto lub, jeśli już je posiadamy, logujemy się na nie. Po zalogowaniu się jest wyświetlana lista utworzonych projektów – mamy też możliwość tworzenia nowych. Będzie to nasz pierwszy projekt, więc klikamy *Create New Project* (**rysunek 2**). Nadajemy nazwę naszemu projektowi, wybieramy platformę sprzętową oraz to, w jaki sposób łączymy się z siecią, a następnie *Create* (**rysunek 3**).

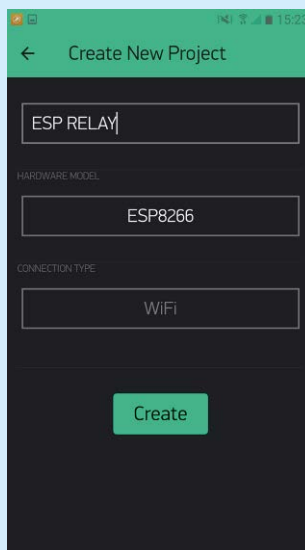
Otworzy się główne okno robocze projektu pokazane na **rysunku 4**. W lewym górnym rogu mamy ikonkę powrotu. W prawym górnym kolejno: ustawienia (ikonka nakrętki), dodanie nowej kontrolki (ikonka plusa w kółeczku) poprzez *Widget Box* oraz uruchomienie (znaczek play). Poniżej widnieje przestrzeń robocza – to tu będziemy układać potrzebne nam komponenty.



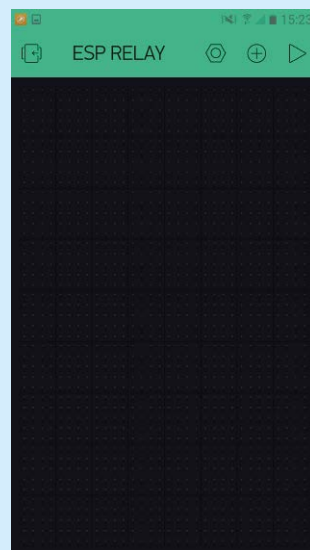
Rysunek 1. Ekran powitalny platformy Blynk



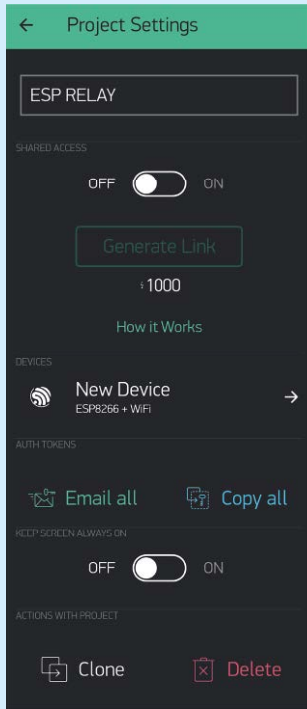
Rysunek 2. Tworzenie nowego projektu



Rysunek 3. Tworzenie połączenia z siecią



Rysunek 4. Główne okno robocze projektu



Rysunek 5. Ustawienia projektu

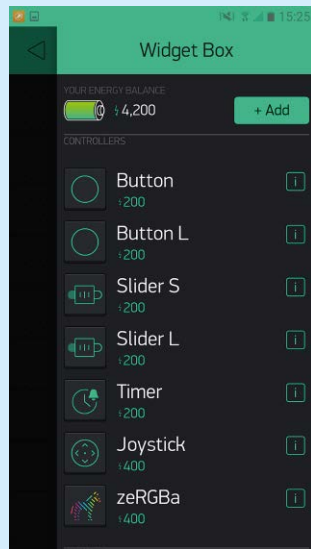
możliwość wygenerowania linku do tego celu. Kolejną opcją jest zmiana platformy sprzętowej. Poniżej możemy wysłać na e-mail podany przy zalogowaniu klucz autoryzacyjny. Każdy projekt ma swój unikalny kod, który musimy także podać w programie – to z jego użyciem aplikacja komunikuje się z urządzeniem. Ten klucz jest jednocześnie zabezpieczeniem, że nikt nieuprawniony nie będzie wysyłał komunikatów w naszym imieniu. Mamy też możliwość ustawienia, aby ekran telefonu/tabletu nie wyłączał się automatycznie. Daje to możliwość zamontowania tabletu w łatwo dostępnym miejscu, jako panel sterowania inteligentnym domem. Ostatnie opcje to klonowanie projektu na inne urządzenie oraz możliwość jego usunięcia.

Za pomocą ikonki strzałki w lewym górnym rogu cofamy się do przestrzeni roboczej. Możemy teraz wcisnąć *Play*, ale po chwili otrzymamy komunikat „Device is offline” (urządzenie jest odłączone), co oczywiście jest prawdą. Zostanie wyświetlona ikona *Stop* – klikamy na nią. Teraz sprawdzimy, co znajduje się w *Widget Box*.

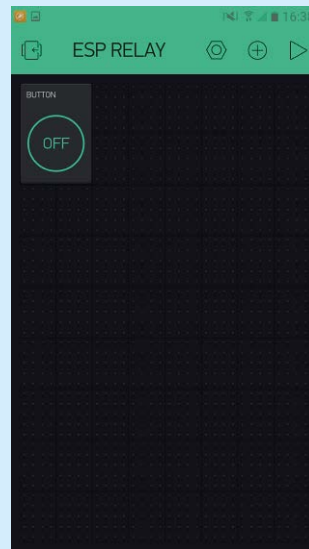
Na górze okna (**rysunek 6**) widzimy naszą „energię”. Blynk niestety nie jest całkowicie darmową aplikacją, każdy komponent kosztuje nas określoną ilość „energii”. Po jej wyczerpaniu nie możemy dodać ich więcej, a „energię” można dokupić. Nie znika ona jednak po skasowaniu kontrolki, a jest oddawana. Domyślnie mamy 4200 jednostek energii – wystarczy to w zupełności do średnio skomplikowanych projektów i żeby zapoznać się z platformą. Poniżej widzimy dostępne kontrolki wraz z ceną energii za ich użycie i ikonką informacji – po jej wciśnięciu wyświetli się dokładna instrukcja użytkownika wraz z przykładowymi kodami itp. Opis warty uwagi komponentów znajduje się na końcu artykułu.

Ustalmy teraz, jaki cel chcemy osiągnąć – w ESP RELAY mamy dostępne 4 przekaźniki, więc utwórzmy 4 przyciski do sterowania nimi. Z *Widget Boxa* wybieramy *Button* – jest on wyświetlany na obszarze roboczym, jak pokazano na **rysunku 7**. Klikamy w niego, aby przejść do ustawień – widzimy je na **rysunku 8**. Możemy:

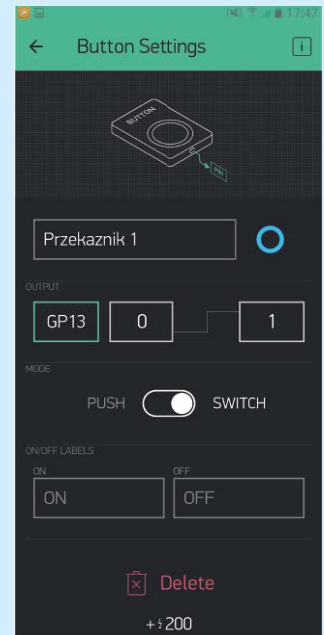
- Ustawić nazwę przycisku oraz jego kolor.
- Przypisać, czym ma on sterować, klikamy *PIN* i otwiera się nowe menu, jak na **rysunku 9**. Tutaj mamy 3 główne kategorie: 1. Digital – sterowanie wyprowadzeniami cyfrowymi.



Rysunek 6. „Energia” projektu



Rysunek 7. Wybranie komponentu Button



Rysunek 8. Ekran ustawień komponentu

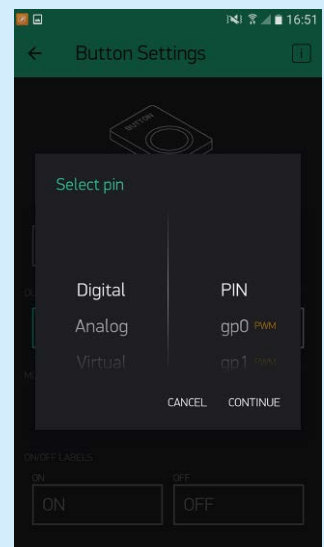
2. Analog – osobna kategoria na wyprowadzenia, które mają także ADC.
 3. Virtual to zmienne, które możemy wykorzystywać w programie.
- Wybrać tryb przycisku – czy ma być chwilowy (zaświeca się tylko, gdy go trzymamy), czy bistabilny – kliknięcie oznacza zmianę stanu na przeciwny.
 - Ustawić etykiety, które będą się wyświetlały w zależności od stanu przycisku.
 - Skasować komponent i odzyskać dzięki temu energię.

W ESP RELAY korzystamy kolejno z wyprowadzeń o numerach: 13, 14, 5, 4. Ustawienia pierwszego przycisku wykonujemy w sposób pokazany na rysunku 8. Resztę analogicznie, ale z innymi wyprowadzeniami. Warto zauważyć, że Blynk nie pozwoli nam na przypisanie jednego doprowadzenia I/O do kilku przycisków – te zajęte zostaną oznaczone będą „busy” (zajęte), jak pokazano na **rysunku 10**. Po dodaniu przycisków dla wszystkich kanałów i ich przemieszczeniu otrzymamy efekt pokazany na **rysunku 11**.

Powróćmy teraz do kwestii sprzętowej. Do Arduino IDE musimy dodać bibliotekę obsługującą Blynka. Pobieramy ją ze strony <http://www.blynk.cc/getting-started/> i kopiujemy do folderu głównego środowiska. Po uruchomieniu Arduino IDE wchodzimy w *Plik* → *Przykłady* → *Blynk* → *Boards Wi-Fi* → *ESP8266 Standalone*. W projekcie musimy jedynie podać 3 parametry:

1. Nazwę sieci, z którą ma się połączyć nasz moduł.
2. Hasło do sieci Wi-Fi.
3. Token autoryzacyjny (wygenerowany przez Blynk).

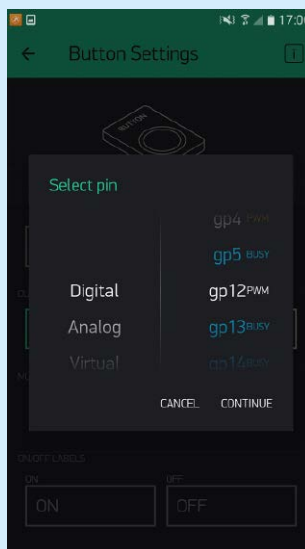
Nie pozostaje nam nic innego, jak wgrać program na urządzenie, pamiętając, że trzeba je wcześniej wprowadzić w tryb programowania. Nie trzeba nawet ustawiać w kodzie pinów jako wyjścia – tym wszystkim zajmą się funkcje z biblioteki Blynka po połączeniu się z aplikacją.



Rysunek 9. Menu PIN

Ostatnim krokiem jest uruchomienie aplikacji na telefonie przyciskiem *Play* – tym razem powinno nastąpić udane połączenie i możemy sterować przekaźnikami przez Internet. Zauważmy, że praktycznie nie napisaliśmy żadnego kodu do tego celu – wszystko zrealizowaliśmy z poziomu Blynka! Wykorzystaliśmy jedynie ułamek możliwości platformy. Pozostałe komponenty, o których warto wspomnieć wraz z przykładowymi zastosowaniami, to:

- **Slider** (suwak), za którego pomocą możemy sterować bezpośrednio wypełnieniem PWM wyprowadzenia, np. do ustawiania jasności lamp LED.
- **Timer**, dzięki któremu możemy ustawić włączenie/wyłączenie pinu o zadanej przez nas godzinie, np. do włączenia podświetlenia podwórza w nocy.
- **Joystick** to dwa połączone suwaki w formie analogowej gałki z kontrolerów do gier, np. do sterowania robotem.
- **zeRGBra** jest obrazkiem zebry, na którym możemy ustawić wybrany kolor, np. do sterowania podświetleniem lamp RGB.
- **Value Display** to wyświetlacz wartości z urządzenia, np. do pokazywania temperatury.
- **LED** to niewielki komponent symulujący diodę świecącą, np. do sygnalizowania napięcia występującego na wejściu.
- **Gauge** jest wskaźnikiem analogowym, np. do wyświetlenia temperatury.
- **LCD** to wirtualny wyświetlacz, na którym możemy pokazywać komunikaty tekstowe.
- **Graph** służący do rysowania wykresów.
- **Video Streaming** mogący posłużyć do podglądu obrazu z kamer internetowych.
- **Twitter** umożliwia wysyłanie powiadomień na Twittera.
- **Notification** umożliwia wyświetlanie powiadomień na telefonie, na przykład alarmów.



Rysunek 10. Oznaczenie używanych przycisków



Rysunek 11. Ekran aplikacji sterującej

- **Email** służy do wysyłania wiadomości e-mail.
- **GPS Trigger**, za którego pomocą możemy stwierdzić, czy nasz telefon znajduje się w określonej lokalizacji, na przykład, gdy nie ma nas w domu, włączy światło, symulując obecność domowników.

Zachęcam do zapoznania się ze wszystkimi możliwościami Blynka. Chciałbym jeszcze zaznaczyć, że projekt jest cały czas rozwijany i ilustracje z artykułu mogą po czasie odbiegać od rzeczywistości.

Przemek Michalak

www.projektydmb.blogspot.com

REKLAMA

Wszystko, co lubisz,
w jednym miejscu



UlubionyKiosk.pl

Oferuje papierowe i elektroniczne wydania czasopism z najważniejszych segmentów rynku:

budownictwo i wnętrza, muzyka i dźwięk, elektronika i automatyka, edukacja i hi-tech, rodzina.

Przesyłka
GRATIS