

<b>Politechnika Świętokrzyska</b> <b>Wydział Elektrotechniki, Automatyki i Informatyki</b>
Technologie IoT rozproszone sieci sensoryczne
Laboratorium 1
Adrian Dorota 3ID15B

## 1. GitHub

### Zastosowanie i opis systemu kontroli wersji.

**GitHub** – hostingowy serwis internetowy przeznaczony dla projektów programistycznych wykorzystujących system kontroli wersji Git. Stworzony został przy wykorzystaniu frameworka Ruby on Rails i języka Erlang. Serwis działa od kwietnia 2008 roku. W kwietniu 2011 ogłoszono, iż GitHub obsługuje 2 miliony repozytoriów. GitHub udostępnia darmowy hosting programów open source oraz płatne prywatne repozytoria. W czerwcu 2018 ogłoszono, iż serwis zostanie przejęty przez przedsiębiorstwo Microsoft za kwotę 7,5 miliarda dolarów.

Kilka z funkcji, które występują w systemie GitHub:

- **bugtracker**,
- **forki repozytoriów**, czyli kopia w osobnym repozytorium należąca do innego użytkownika,
- **pull requesty** – osoba mająca forka może zgłosić swój kod do złączenia z głównym repozytorium,
- **statystyki**,
- **organizacje zrzeszające programistów pracujących nad repozytoriami**,

- **web hooks** – wywołanie operacji na repozytorium wysyła informację do innego serwisu lub skryptu,
- **wiki** dla celów dokumentacji.

Ponadto GitHub udostępnia usługę zwaną GitHub Pages, służącą do szybkiego tworzenia stron internetowych, których kod jest opublikowany w ramach repozytoriów Gita umieszczonych w serwisie.

Możliwości usługi:

- darmowe stworzenie strony personalnej, dla organizacji lub projektu,
- możliwość utworzenia własnej strony w oparciu o statyczne pliki strony internetowej lub stworzenie bloga przy zastosowaniu silnika blogowego Jekyll,
- możliwość podłączenia własnej domeny do danej strony internetowej (lub skorzystania z darmowej subdomeny w ramach adresu github.io),
- darmowy certyfikat SSL dla wszystkich stron w domenie github.io, z możliwością wygenerowania certyfikatu od Let's Encrypt dla stron obsługujących własne domeny.

## Podstawowe komendy

**git init** – Inicjalizuje repozytorium GIT w danym katalogu

**git add [nazwa\_pliku]** – Dodaje zmiany we wskazanym pliku do commita

**git add** – Dodaje wszystkie zmienione pliki do commita

**git add -p [nazwa\_pliku]** – Udostępnia możliwość dodania wybranych linii w zmodyfikowanym pliku do commita

**git commit -m "[treść\_commita]"** – Dodaje opis do commita. Dobrym zwyczajem jest opisanie co ta zmiana wprowadza do kodu w zakresie funkcjonalnym

**git add origin [adres\_repozytorium, np. <https://github.com/username/ moje-repozytorium.git>]** – Ustawia konkretny adres zdalnego repozytorium jako główne repozytorium

**git push origin master** – Wysłanie zmian do branacha zdalnego

**git push -f** – Wysłanie zmian do zdalnego repozytorium ignorując konflikty, to znaczy, że jeśli wystąpią konflikty to pliki zostaną nadpisane właśnie wysłaną wersją. Trzeba stosować to bardzo ostrożnie.

***git checkout [nazwa\_brancha]*** – Zmienia aktywny branch na wybrany przez użytkownika

***git checkout [nazwa\_pliku]*** – Usuwa zmiany w wybranym pliku

***git checkout*** – Usuwa zmiany we wszystkich zmienionych plikach

***git checkout -b [nazwa\_brancha]*** – Tworzenie nowego brancha z aktywnego brancha i przełączenie się na niego

***git rebase master*** – Zaciągnięcie zmian z brancha głównego do brancha aktywnego

***git push origin :[nazwa\_brancha]*** – Usunięcie zdalnego brancha

***git branch -d [nazwa\_brancha]*** – Usuwanie brancha lokalnie. Nie można usunąć w ten sposób aktywnego brancha

***git stash*** – Dodanie zmienionych plików do pamięci/stosu i usunięcie ich z aktywnego brancha

***git pull --rebase*** – Pobranie najnowszych zmian z aktywnego brancha zdalnego

***git stash pop*** – Przywrócenie zmodyfikowanych plików z pamięci/stosu

***git stash clear*** – Czyszczenie pamięci/stosu

***git remote prune origin*** – Pobranie aktualizacji o usuniętych branchach zdalnych

***git fetch --all*** – Pobranie listy zdalnych branchy

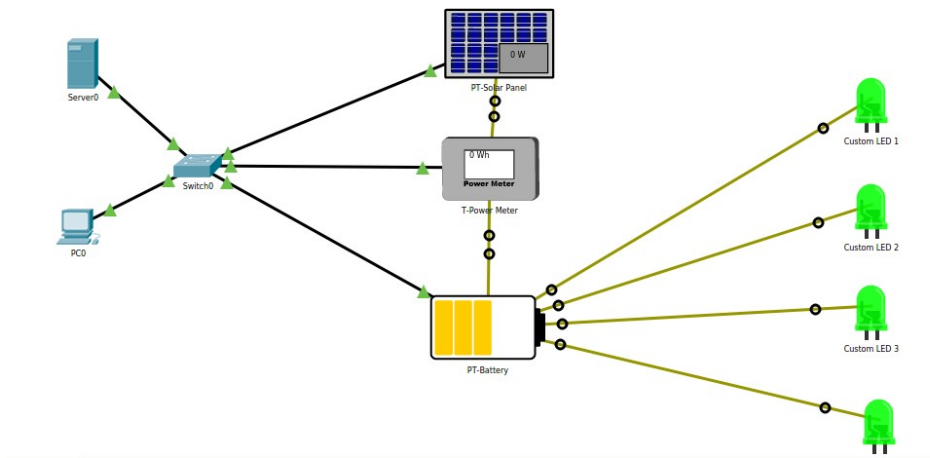
***git branch*** – Wyświetlenie listy lokalnych branchy

***git branch -r*** – Wyświetlenie listy zdalnych branchy

***git status*** – Wyświetlenie listy zmienionych plików

***git diff [nazwa\_pliku]*** – Szczegółowe wyświetlenie zmian w wybranym pliku

## 2. Część obserwacyjna z laboratorium



A) Ogniwo fotowoltaiczne przetwarza światło zdefiniowane w opcjach środowiskowych na prąd elektryczny. Wartość mocy wytwóźnego prądu jest przedstawiona na „wyświetlaczu” ogniwa.

Dalej jest przekazywany do miernika, który, co ciekawe pokazuje wartość mniejszą niż na wyświetlaczu ogniwa i przekazywany do baterii ładując ją.

Wartości te można podejrzeć także logując się na serwer w podpunkcie B.

B)

The screenshot shows the 'IoT Server - Devices' web interface. The interface has a navigation bar with 'Physical', 'Config', 'Desktop' (selected), 'Programming', and 'Attributes'. Below the navigation bar, there is a 'Web Browser' window showing the URL 'http://1.0.0.1/home.html'. The main content area displays a table of devices:

Device ID	Device Name	Device Type	Status
IoT0 (PTT08102U0T-)	Battery	Available power	0.02 %
IoT2 (PTT0810VKSK-)	Power Meter	Status	1 Watts
IoT1 (PTT08103V12-)	Solar	Status	0 Wh

At the bottom of the interface, there is a 'Top' button.

C) Po odłączeniu fotowoltaiki LEDy pobierają energię z baterii do momentu całkowitego jej rozładowania. Czas działania jest uzależniony od pierwotnego poziomu naładowania baterii oraz ilości LEDów naraz podłączonych do baterii.

D) Działanie urządzeń:

A) ogniwo fotowoltaiczne - Ogniwo fotowoltaiczne przetwarza światło zdefiniowane w opcjach środowiskowych na prąd elektryczny. Wyjście wykorzystuje skalę logarytmiczną w celu dostosowania wartości mocy do 100 MW(+/-20%)

A) Wyjście D0 - przekazuje prąd na linię.

B) miernik mocy – Odczytuje prąd który płynie w obwodzie.

Wejście i wyjście wykorzystuje skalę logarytmiczną w celu dostosowania wartości mocy do 100 MW(+/-20%).

A) Wejście D0 – czyta prąd od innego urządzenia

B) Wyjście D1 – przekazuje prąd dalej.

C) Standardowy LED zielony – świeci na zielono gdy ma dostęp do prądu.

Poziom świecenia jest uzależniony od wartości na wejściu, najjaśniejszy przy 2023, wyłączony przy 0. Przyjmuje wartości 0-1023.

### **Możliwości rozbudowy:**

1. Zdalne sterowanie LEDami by świeciły na żądanie przez aplikację mobilną , zastosowanie włącznika ściennego, bądź czujnik ruchu (oświetlenie klatki schodowej)
2. Zastosowanie zegara czasu rzeczywistego by układ świecił tylko w określonych godzinach przez określony czas (oświetlenie podjazdu w nocy)