

DOKUMENTACJA

Nazwa zespołu : Gwardia Złotego Druta

Skład zespołu:

Mikołaj Barcikowski (Kapitan)

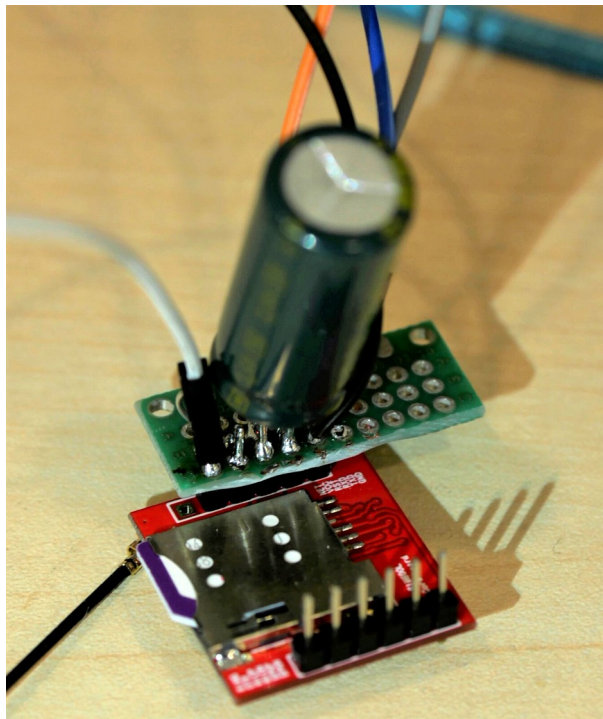
Paweł Wojciechowski

Luiza Błaszczak

Jakub Chmielewski

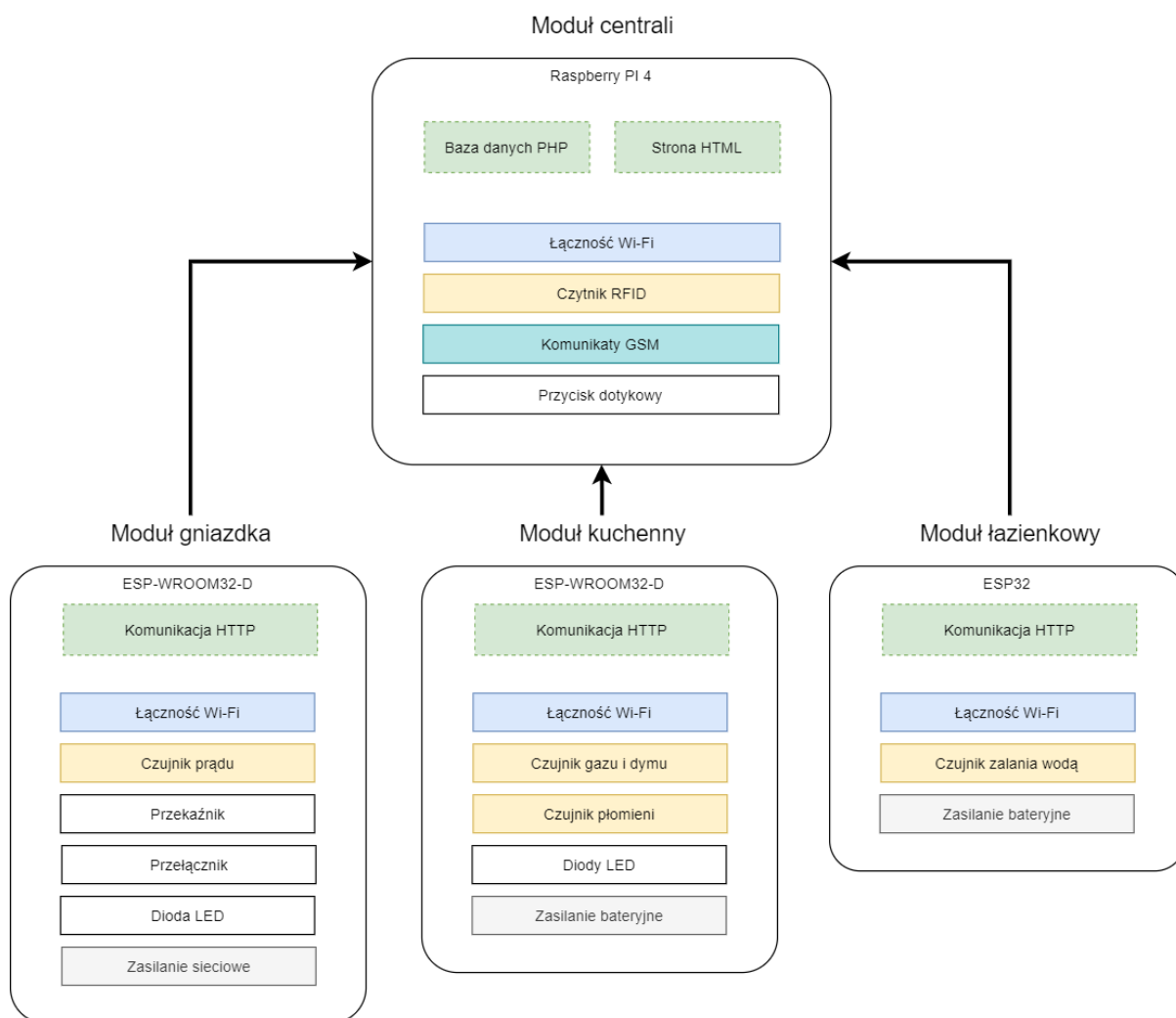
Hardware (Paweł i Jakub)

Głównym elementem naszego systemu jest dostarczony od zespołu BHL, minikomputer Raspberry Pi model 4. Dla naszych potrzeb został on skonfigurowany w postaci serwera na którym wyświetlana jest strona internetowa w formacie html umożliwiająca zarządzanie systemem, podgląd parametrów i jego konfigurację. Dostęp do strony jest realizowany przez panel logowania zapewniający walidację osoby logującej. Serwer współpracuje z bazą danych. Do Raspberry Pi zostały dołączone 3 moduły, są to kolejno: moduł GSM SIM800L, czujnik RFID RC522 oraz czujnik dotyku TTP223B. Moduł GSM umożliwia połączenie się do sieci komórkowej, a co za tym idzie obsługę wiadomości tekstowych. Zostały one wykorzystane do informowania użytkownika o występujących problemach, bez potrzeby wykorzystania np. zewnętrznego serwera. Czujnik RFID i czujnik dotyku zostały wykorzystane na potrzeby załączania i wyłączania systemu komunikacji.



Moduł GSM SIM800L podłączony do Raspberry Pi

Sub Modułami naszego systemu są płytki ESP-WROOM32-D realizujące węzły pomiarowe o różnicowanym zastosowaniu. Komunikacja z bazą danych następuje po interfejsie WiFi oraz z wykorzystaniem protokołu HTTP. Każdy węzeł pomiarowy poza komunikacją bezprzewodową z bazą danych posiada unikalne dla niego sensory, dzięki czemu zaprezentowano możliwość dołączenia węzłów w różnych miejscach i otoczeniu.



Gniazdko domowe

Pierwszym z węzłów pomiarowych stanowi moduł gniazdka sieciowego wyposażonego w czujnik prądu ACS712, umożliwiający pomiar prądu zarówno stałego oraz zmiennego w zakresie od 0 do 5A co w zupełności wystarcza dla zdecydowanej większości asortymentu domowego. Ponadto zaimplementowano moduł przełącznika, który umożliwia zdalne kontrolowanie pracy gniazdka (jego włączenie i wyłączenie poprzez stronę internetową). W przypadku wykrycia uruchomienia nowego urządzenia do gniazdka (ustawienia domyślne zakładają pobór prądu powyżej 0,5A lecz możliwe jest ustawienie pożądanej wartości), urządzenie transmituje informacje o poborze energii na stronę internetową. W przypadku gdy gniazdko zostanie wyłączone

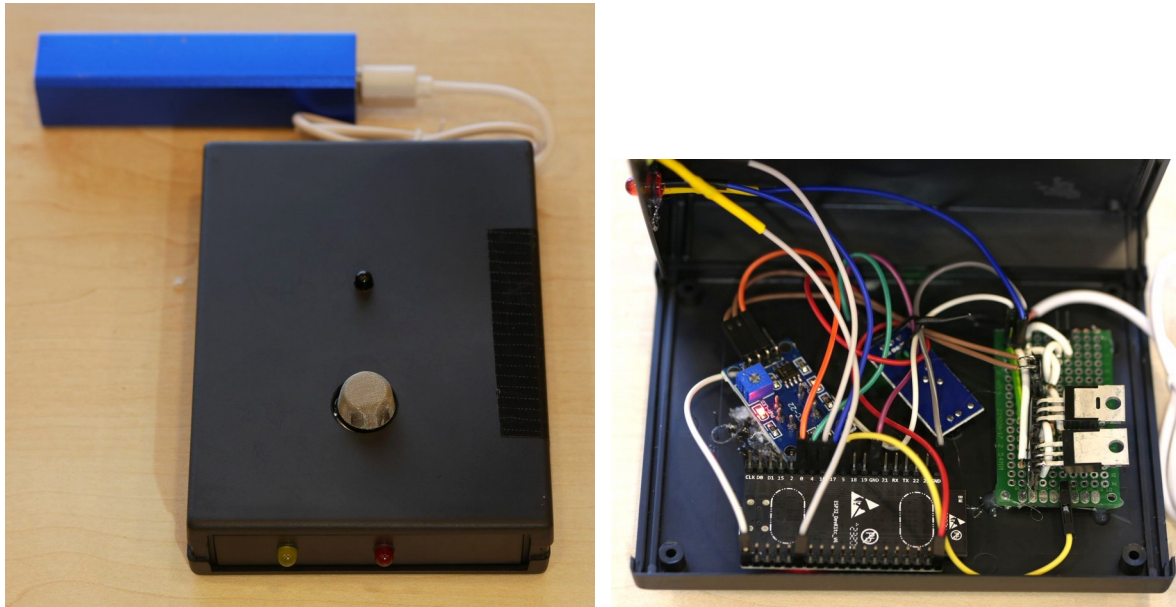
zdalnie, możliwe jest ręczne jego uruchomienie poprzez przycisk znajdujący się na jego obudowie. Do zasilenia elementów wykorzystano przetwornicę transformującą napięcie sieciowe na 5V DC.



Panel przedni oraz panel tylni modułu gniazdka

Kontrola zanieczyszczeń powietrza

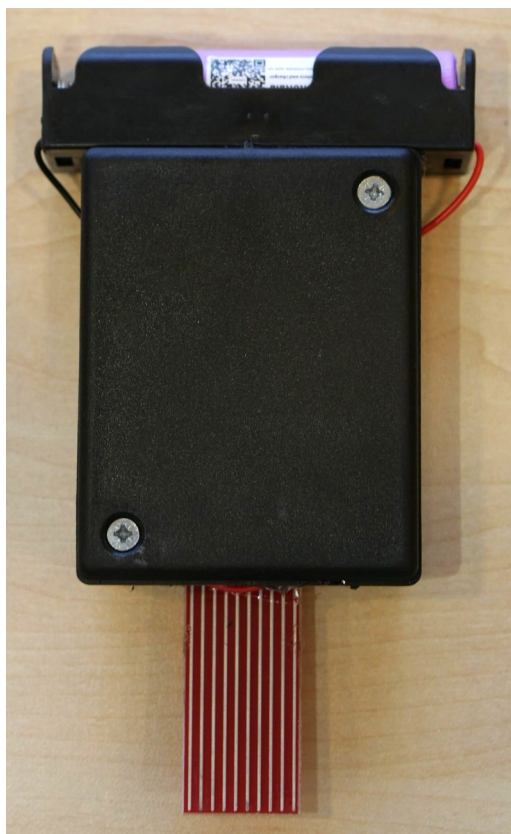
Kolejny moduł wykorzystuje czujnik gazów łatwopalnych i dymu MQ-2, który reaguje na stężenie takich gazów jak propan, butan, metan, H₂ oraz alkohol. Współpracuje on z czujnikiem ognia. Zarówno jeden jak i drugi czujnik, w przypadku detekcji przekazuje informacje na stronę internetową. Warto wspomnieć, że z uwagi na wysoki pobór energii przez pierwszy czujników, zastosowano parę nmos+pmos podłączoną do jednego z pinów GPIO celem umożliwienia ograniczenia jego aktywności wyłącznie do okresowego działania. Do zasilenia tego modułu wykorzystano zasilanie bateryjne.



Obudowa i wnętrze modułu kuchennego

Kontrola zalańia

Moduł z czujnikiem zalania sprawdza czy w pomieszczeniu w którym się znajduje nie doszło, nie dochodzi do zalania/podtopień. Podobnie jak w poprzednim module, wykorzystano zasilanie bateryjne.



Moduł łazienkowy (kontrola zalania)

Firmware (Paweł)

Oprogramowanie do każdego z modułów ESP32 zostało wykonane w języku C. Środowisko jakie wykorzystano to Code::Blocks oraz narzędzie Cmake do generowania plików projektowych.

W urządzeniach wykorzystano system operacyjny czasu rzeczywistego FreeRTOS, który wspierał wykonywanie wielu operacji z wykorzystaniem przydzielonych zadań np. wysyłanie informacji przez protokół HTTP, obsługa czujników.

Każdy z modułów swoją pracę rozpoczyna od zainicjalizowania swoich podstawowych parametrów i wyjść/wejść. Następnie, przechodzi do nawiązywania połączenia WiFi (należy wspomnieć, że w dowolnym momencie gdy moduł utraci połączenie, będzie próbował cały czas je przywrócić). Następnie z wykorzystaniem protokołu SNTP pobiera aktualny czas. Finalnie przechodzi do realizacji zadań dedykowanych dla konkretnego modułu.

W przypadku modułu gniazdka sieciowego, skorzystano z formatu JSON celem odczytu informacji z bazy danych, z tego względu skorzystano z biblioteki cJSON.

Niektóre czujniki wymagały odczytu z przetwornika ADC. Z tego względu zrealizowano uśrednianie pomiarów celem najwierniejszego ich odwzorowania.

Aplikacja webowa (Luiza)

Do przygotowania aplikacji webowej wykorzystano język php oraz HTML, wraz z bazą danych mysql. Serwer został postawiony na Raspberry Pi 4, która w naszym projekcie była głównym modułem zarządzającym systemem zabezpieczeń. Urządzenia wchodzące w skład systemu zostały połączone z siecią Wi-Fi, dzięki czemu uzyskano dostęp do strony z wszystkich urządzeń w sieci. W dalszym etapie rozwoju projektu baza zostałaby umieszczona na serwerze zewnętrznym, tak aby umożliwić do niej dostęp spoza sieci lokalnej.

Po wpisaniu w pasek przeglądarki adresu 192.168.0.102/strona/index.php pojawia się strona logowania.

Login:

Hasło:

Zaloguj się

Wprowadzane dane są zabezpieczone przed zjawiskiem “wstrzykiwania sql” oraz rozróżniane są wielkie i małe litery. Błędne wpisanie hasła powoduje wyświetlenie komunikatu “Nieprawidłowy login lub hasło!”.

Po zalogowaniu zostaje uruchomiona sesja użytkownika (po 1440 sekundach bezczynności nastąpi automatyczne wylogowanie).

Witaj admin! [Wyloguj się!](#)

Numer gniazdka	Status	Sterowanie
1	0	Włącz gniazdko

Gniazdko nr 1

Sesja pomiarowa	Zużycie energii [kWh]
1	2.193058570375E-5
2	0.01217449213361
3	0.0064594955083881

Sensor płomieni nr 1 wykrył ogień!

Sensor 1 wykrył dym!

Sensor wody nr 1 wykrył powódź!

Na stronie są wyświetlane wszystkie gniazdka wyposażone w sterowniki na ESP32. Gdy czujnik natężenia prądu wykryje wartość większą niż 0,5A to do bazy zaczynają przychodzić informacje o czasie wykonania pomiaru, wartości natężenia prądu, ID gniazdka oraz sesji pomiarowej. Sesje pomiarowe są wykorzystywane do ustalenia, ile energii zostało pobrane w danym przedziale czasowym i wyznaczenia kWh. Z poziomu strony możemy włączać oraz wyłączać gniazdka (zabezpieczenie dla osób zapominalskich). Na stronie pojawiają się również informacje o wykryciu dymu oraz ognia, a także informacje o zalaniu pomieszczenia. Każdy z komunikatów można anulować,

jednocześnie usuwając rekordy z bazy danych.

<input type="checkbox"/>	bathroom_water	★
<input type="checkbox"/>	kitchen_fire	★
<input type="checkbox"/>	kitchen_smoke	★
<input type="checkbox"/>	socket_data	★
<input type="checkbox"/>	socket_status	★
<input type="checkbox"/>	users	★
6 tabele		Sum

Baza danych zawiera 6 tabel pogrupowanych tematycznie:

- bathroom_water - odpowiada za gromadzenie informacji o wykryciu zalania pomieszczenia,
- kitchen_fire - odbiera informacje o wykryciu pożaru w kuchni,
- kitchen_smoke - wykrycie dymu w kuchni,
- socket_data - pobieranie danych o poborze prądu z gniazdek,
- socket_status - przechowuje stan gniazdek - włączone/wyłączone,
- users - tabela nazw użytkowników, haseł oraz telefonów do wysyłania wiadomości SMS z informacjami o niebezpieczeństwach wykrytych w budynku.

Security (Mikołaj)

Raspberry Pi wraz z przyłączonymi modułami pozwolił na łatwe i przyjazne użytkownikowi sterowanie zabezpieczeniami. Do uruchamiania systemu powiadomień wykorzystano czujnik dotyku, który nie wymaga skomplikowanych czynności, co jest korzystne z punktu widzenia użytkownika podczas wychodzenia z domu czy mieszkania. Do odblokowania systemu wykorzystano moduł RFID, który wykorzystuje unikalne tagi dla każdego z zarejestrowanych użytkowników. Takie zabezpieczenie pozwala myśleć np. o doposażeniu systemu w czujki PIR lub czujniki mikrofalowe do wykrywania ruchu.

Wykorzystany moduł GSM, wraz z oprogramowaniem w języku Python, pozwala na wysyłanie SMS-ów w prosty sposób za pomocą komend AT. SMS-y są wysyłane automatycznie i treści zostały skonfigurowane pod możliwe do wystąpienia zdarzenia. Ze względu na bardzo duży szczytowy pobór prądu (sięgający 2A według dokumentacji) wystąpiła potrzeba zastosowania kondensatora elektrolitycznego o dużej pojemności, w tym wypadku 1000uF. Układ ten stanowił problem nawet dla zasilacza

laboratoryjnego. Aby wysyłać SMS-y o dostosowanej treści, z bazy danych pobierano potrzebne dane dzięki możliwościom języka Python. W zależności od zdarzenia, pobierano dane świadczące o włączeniu/wyłączeniu danego urządzenia(gniazdko) lub dane o wystąpieniu danego zdarzenia w bazie. Aby umożliwić jednoczesne oczekiwanie na wystąpienia zdarzenia oraz oczekiwanie na osobę z tagiem RFID, wykorzystano wątki. Do podłączenia wymienionych modułów wykorzystano różne interfejsy, takie jak SPI, UART czy po prostu wyjścia cyfrowe.