WYŻSZA SZKOŁA BANKOWA W POZNANIU

Wydział Finansów i Bankowości

Łukasz Piasecki

Projekt systemu nadzoru i sterowania dla wybranych systemów bytowych przedsiębiorstwa.

Praca magisterska

Promotor

Dr hab. Wojciech Rudziński

Poznań 2021

Spis treści

Rozdział 1
Wstęp

Tematyka inteligentnych budynków w ostatnich latach budzi spore zainteresowanie. Trend ten widoczny jest zarówno w budownictwie mieszkaniowym jak i w przemyśle. Wpływ na to mają rosnąca ilość dostępnej w Internecie wiedzy na ten temat oraz coraz bardziej dostępna cena podzespołów niezbędnych do realizacji różnych projektów Internetu Rzeczy.

Rozdział 2
Założenia teoretyczne

2.1 Nadzór i sterownie

Systemy nadzoru i sterowania komunikują się z czujnikami i urządzeniami wykonawczymi, aby dostarczyć informacji o bieżącym stanie obiektu (nadzór) lub dokonać zmiany tego stanu (sterowanie). Czynności te mogą dokonywać się w czasie rzeczywistym, w określonych chwilach lub tylko w reakcji na określone zdarzenie. Obie te operacje można wykonywać lokalnie, przy bezpośrednim dostępie do podzespołów spełniających te funkcje lub zdalnie, nawet na innym kontynencie.

2.2 Systemy bytowe w ogólności

System bytowy to taki, który jest związany z działaniem lub funkcjonowaniem człowieka, przedsiębiorstwa lub procesu. Zaliczyć do nich można:

• Systemy kontroli dostępu

• Systemy powiadamiania o pożarze oraz wczesnej detekcji dymu

• Systemy antywłamaniowe oraz alarmowe

• Systemy ochrony przed zalaniem, zaczadzeniem lub zagazowaniem

Każdy z wyżej wymienionych zostanie krótko scharakteryzowany w dalszej części pracy.

2.3 Kontrola dostępu

Systemami kontroli dostępu nazywamy oprogramowanie oraz urządzenia pozwalające na ograniczenie (kontrolę) dostępu osób do kontrolowanego obiektu. Bardzo częstym jest tutaj podział obiektu na mniejsze części (strefy) i zróżnicowanie użytkowników systemu lub ich grup pod względem możliwości dostępu do tych stref. W rozwiązaniach bardziej zaawansowanych możliwe jest też ustalenie dni i godzin, w których wstęp jest dozwolony lub zabroniony poszczególnym grupom, zdefiniowanie dni świątecznych w danym roku kalendarzowym, kontrola ilości osób w danej strefie (np. za pomocą kołowrotu), tryby wejścia komisyjnego (konieczna autoryzacja dwóch lub więcej uprawnionych użytkowników) oraz specjalny rodzaj wejścia tzw. wymuszony lub pod przymusem. Polega on na tym na takim sposobie uzyskania dostępu (innym kodem, naciskając ukryty

przycisk, przykładając kartę na dłużej lub dwukrotnie) który jednoczenie wysyła do systemu informację, że użytkownik został zmuszony do tych działań i może być w niebezpieczeństwie oraz potrzebować pomocy. Tryb ten można spotkać nie tylko w sejfach i kasach pancernych, lecz także w przemysłowych systemach kontroli dostępu do budynków. Do uwierzytelnienia w tego typu systemach można użyć zarówno przedmiotów takich jak karty lub breloki w technologii RFID (rzadziej NFC), danych biometrycznych jak odciski linii papilarnych, skan siatkówki oka lub geometria twarzy, jak i pilotów radiowych lub aplikacji w telefonie.

2.4 Zawiadamianie o pożarze

Systemami zawiadamiania o pożarze nazywamy zespoły urządzeń oraz programów umożliwiających automatyczną detekcję pożaru (czujniki dymu i temperatury), proste i szybkie ręczne informowanie o pożarze wykrytym przez człowieka (przyciski ręcznego ostrzegania pożarowego tzw. ROP) oraz alarmowanie o wykrytym zagrożeniu (sygnalizatory akustyczne i wizualne, moduły komunikacji przewodowej i bezprzewodowej wysyłające informację o pożarze do wskazanych odbiorców). Urządzenia te powinny cechować się wysoką sprawnością i niezawodnością, min. spełniać normy określające, ile czasu dane urządzenie powinno wytrzymać przy ekspozycji na otwarty ogień.

2.5 System antywłamaniowy

System antywłamaniowy służy do ochrony mienia przed grabieżą, zniszczeniem oraz włamaniem. Podobnie jak opisany wcześniej system powiadamiania o pożarze podzielić go można na elementy służące do wykrycia zdarzenia, takie jak czujniki ruchu oparte o detekcję promieniowania podczerwonego lub ultradźwiękowe oraz urządzenia sygnalizacyjne zarówno bezpośrednio informujące o włamaniu światłem i dźwiękiem jak i wysyłające komunikaty do zdefiniowanych adresatów np. mailem lub sms-em. Opcjonalnym elementem są podobnie jak w systemach kontroli dostępu komponenty służące do uruchomienia tzw. „cichego alarmu” czyli powiadomienia o niebezpieczeństwie bez informowania o tym osób naruszających strefę, np. w formie przycisku na pilocie, połączenia na odpowiedni numer lub aplikacji w telefonie.

2.6 Czujniki zalania i zadymienia

Systemy ochrony przed zalaniem oraz umożliwiające detekcję gazów łatwopalnych lub niebezpiecznych jak metan lub tlenek węgla zwany potocznie czadem zazwyczaj są

zintegrowane w jedno urządzenie zawierające czujnik oraz sygnalizator, zwykle dźwiękowy. W tego typu systemach bardzo dużą rolę odgrywa czas reakcji zwykle liczony w minutach a działanie ogranicza się w pierwszym rzędzie do natychmiastowego opuszczenia obszaru objętego alarmem. Również ze względu na stosunkowo mały zasięg oddziaływania czynników takich jak zalanie wodą lub zagazowanie w zamkniętych pomieszczeniach lepszym rozwiązaniem wydaje się większa ilość pojedynczych czujników i sygnalizatorów, natomiast centralny system ostrzegania, choć bardzo ważny, to ze względu na wspomnianą konieczność bardzo szybkich działań w obszarze wystąpienia alarmu staje się kwestią drugorzędną.

Rozdział 3
Zastosowane technologie

3.1 Komputer Raspberry Pi 4

Platformą sprzętową na której uruchamiane będzie oprogramowanie sterujące podzespołami systemu jest mikrokomputer Raspberry Pi 4B wyposażony w 4GB pamięci RAM z systemem operacyjnym Raspbian. Projekt Raspberry Pi rozpoczął się w 2012 roku i od początku dedykowany jest dla automatyków, robotyków oraz programistów, zarówno doświadczonych jak i tych dopiero rozpoczynających naukę. Obecnie jest to rozwiązanie powszechnie znane i stosowane zarówno w hobbystycznie realizowanych projektach jak i rozwiązaniach przemysłowych. Znaczny wpływ na popularność rozwiązania ma relatywnie niska cena urządzenia oraz duża ilość ogólnodostępnych materiałów.

Rysunek 3. 1 Komputer Raspberry Pi 4B

Źródło: https://botland.com.pl/

3.2 System Raspbian

System do niedawna zwany Raspbianem, a od niedawna Raspberry Pi OS (The Raspberry Pi Foundation, 2021) to jeden z najczęściej używanych i najbardziej znanych systemów operacyjnych na platformę Raspberry Pi. Jest to system oficjalnie uznany przez The Raspberry Pi Foundation, oparty na Debianie, będącym jedną z dystrybucji Linuxa. Działa w architekturze ARM, wykorzystuje środowisko graficzne LXDE oraz PIXEL.

3.3 GPIO

Wejścia-wyjścia ogólnego przeznaczenia (ang. General Purpose Input Output, GPIO) to zestaw pinów stanowiący interfejs komunikacyjny między komputerem a podłączonymi do niego zewnętrznymi urządzeniami. Raspberry Pi posiada takich pinów 40 i umożliwia pracę z nimi w kilku układach, z których najpopularniejsze to GPIO oraz BOARD.

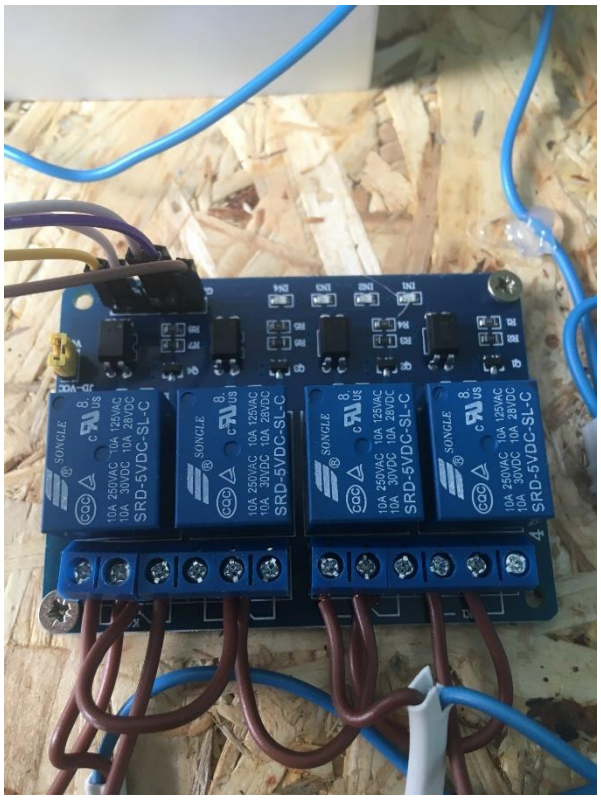
Rysunek 3. 2 Układ pinów GPIO w Raspberry Pi 4B

Źródło: https://pinout.xyz/

3.4 Przekaźnik 5V

Przekaźnik to proste urządzenie elektroniczne, którego zadaniem jest zmiana stanu po wystąpieniu określonych warunków. W projekcie zastosowano układ 4 przekaźników firmy Songle zintegrowanych na jednej płytce sterowanych napięciem 5V DC, gdzie stan niski (logiczne 0) powoduje wyzwolenie przekaźnika. Maksymalny prąd jaki może być podłączany do styków roboczych urządzenia to 10A, o napięciu do 250V dla prądu zmiennego i do 30V dla stałego. W stanie spoczynku styk NC (ang. Normal close) jest zwarty do styku ogólnego, natomiast styk NO (ang. Normal open) jest rozwarty. W przypadku wysterowania przekaźnika, które powodowane jest podaniem na pin sterujący sygnału niskiego następuje odwrócenie sytuacji, tj. styk NC rozwiera się, natomiast styk NO zostaje zwarty do styku ogólnego. Niezwykle prosta zasada działania czyni urządzenie

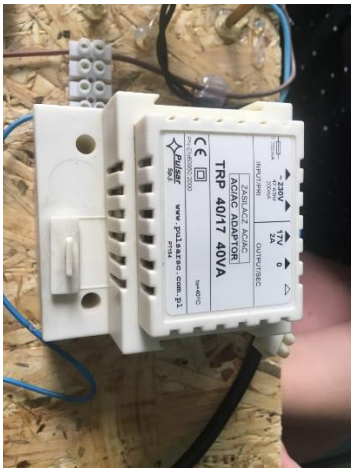
niezwykle wszechstronnym włącznikiem zasilania i umożliwia sterowanie znacznie wyższymi napięciami aniżeli możliwe by to było przy zasilaniu urządzeń bezpośrednio z pinów GPIO Raspberry Pi (max. 5V na stałe, max. 3,3V sterowane).



Rysunek 3. 3 Przekaźnik zastosowany w projekcie

3.5 Zasilacz buforowy 12V

Do zasilania systemu zastosowano zasilacz buforowy 12V firmy Pulsar. Zestaw składa się z transformatora obniżającego napięcie zmienne z sieci do wartości 17V oraz mostka prostującego, na wyjściu, którego pojawia się napięcie stałe o nominalnej wartości 12V. Dodatkowo jest on wyposażony w styki ładowania, umożliwiające podłączenie akumulatora, co pozwala na pracę układu nawet przy zaniku zasilania sieciowego oraz styk TAMPER typu NC, który rozwiera się w przypadku zwolnienia przycisku, który normalnie jest dociskany przez obudowę urządzenia. Pozwala to na wysłanie informacji o otwarciu obudowy, co może ułatwić zapobieganie sabotażowi.



Rysunek 3. 4 Zasilacz obniżający napięcie zmienne do 17V

Rysunek 3. 5 Prostownik zmieniający prąd zmienny o obniżonym napięciu na prąd stały o napięciu nominalnym 12V

3.6 Czytnik RFID RC522

Do obsługi kart oraz breloków działających w technologii RFID zastosowany został czytnik RC522. Urządzenie podłączono do pinów GPIO przy użyciu interfejsu SPI (ang. Serial Peripheral Interface). Pewien problem techniczny stanowił fakt, że urządzenie fabrycznie nie ma przylutowanych kołków goldpin i konieczne jest wykonanie tego we własnym zakresie.

Rysunek 3. 6 Czytnik RFID zastosowany w projekcie

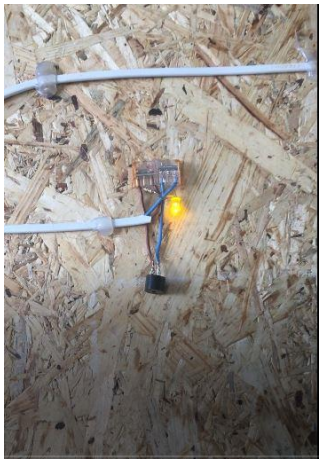
3.7 Kontaktron CMD14

Jako czujnik otwarcia drzwi zastosowano kontaktron CMD14. Jest to urządzenie o bardzo prostej zasadzie działania. Jeden z segmentów kontaktronu jest magnesem. Jak długo urządzenie jest w polu magnetycznym (zwarcie elementów) obwód pozostaje zwarty. Gdy drzwi zostają otwarte, rozwarcie elementów powoduje przerwanie pola magnetycznego, co skutkuje przerwaniem obwodu. Zwarcie bądź rozwarcie obwodu monitorowane jest na jednym z pinów GPIO i dostarcza do systemu informacji o tym czy drzwi są zamknięte.

Rysunek 3. 7 Kontaktron zastosowany w projekcie

3.8 Buzzer

System sygnalizacji o uruchomieniu alarmu stanowi buzzer wydający sygnał modulowany oraz pulsująca dioda LED koloru żółtego. Efekt pulsowania i modulacji dźwięku uzyskano przez cykliczną zmianę stanu przekaźnika, do którego podłączono zarówno diodę jak i buzzer. Jak zostało wspomniane we wcześniejszych rozdziałach napięcie zasilające systemu wynosi 12V, w związku z czym zastosowany został buzzer przystosowany do takiego napięcia, choć na etapie testów stwierdzono, że również urządzenie o nominalnym napięciu 5V działa bezawaryjnie przy zasilaniu 12V.



Rysunek 3. 8 Sygnalizator uruchomienia alarmu składający się z buzzera oraz diody LED

3.9 Czujnik dymu i gazów łatwopalnych

Do detekcji dymu oraz niebezpiecznych gazów zastosowany został czujnik MQ-2. Zgodnie z notą katalogową czujnik zasilany jest napięciem 5V DC i może wykryć stężenia LPG, dymu, alkoholu, propanu, wodoru, metanu i tlenku węgla w zakresie od 200 do 10000 ppm. Czujnik posiada zarówno wyjście cyfrowe jak i analogowe oraz pokrętło umożliwiające skalibrowanie jaka ilość cząsteczek gazu ma uruchomić alarm.

Rysunek 3. 9 Czujnik dymu oraz gazów łatwopalnych zastosowany w projekcie.

3.10 Detektor ruchu PIR HC-SR501

Jako czujnik ruchu wykorzystany został sensor promieniowania podczerwonego HC-SR501. Urządzenie posiada dwa pokrętła umożliwiające kalibrację czułości oraz czasu, przez który musi trwać ruch, aby wyzwolić sygnał na wyjściu czujnika.

Rysunek 3. 10 Czujnik ruchu wykorzystany w projekcie

3.11 Czujnik opadów

Do wykrycia zalania układu wodą lub inną cieczą zastosowany został czujnik o oznaczeniu HL-83, choć nie udało się odnaleźć jego fabrycznej nazwy ani nazwy firmy produkującej go. Urządzenie posiada pokrętło umożliwiające skalibrowanie jaka ilość cieczy na sondzie ma wyzwolić alarm. Podobnie jak czujnik dymu posiada zarówno wyjście cyfrowe jak i analogowe.

Rysunek 3. 11 Sonda czujnika zalania.

Rysunek 3. 12 Czujnik zalania

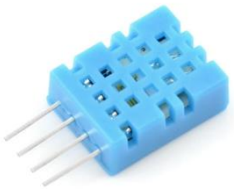
3.12 Wyświetlacz LCD

Do przekazywania informacji użytkownikowi końcowemu służył będzie wyświetlacz LCD 2 x 16 podłączony do Raspberry interfejsem I2C.

Rysunek 3. 13 Wyświetlacz LCD zastosowany w projekcie

3.13 Czujnik temperatury i wilgotności DHT11

Jako termometr i czujnik wilgotności powietrza zastosowano czujnik DHT11 powszechnie używany w prostych projektach z zakresu Internetu rzeczy.



Rysunek 3. 14 Czujnik DHT11
Źródło: https://botland.com.pl/

Rozdział 4
Praktyczna realizacja projektu

4.1 Montaż i łączenie elementów projektu

Układ został zmontowany na płycie wiórowej OSB, poszczególne elementy przytwierdzone zostały mechanicznie za pomocą wkrętów oraz kleju. Połączenia elektronicznie wykonano w kilku różnych technologiach. Były to połączenia lutowane, przy użyciu kostek wago oraz tradycyjnych zaciskowych a także (głównie do pinów GPIO w Raspberry) za pomocą kołków goldpin. Z uwagi na tymczasowy charakter instalacji, będącej projektem badawczym zastosowano powyższe rozwiązania, mając na uwadze niską cenę oraz ogólną dostępność komponentów potrzebnych do montażu. W przypadku zastosowania rozwiązania produkcyjnie konieczna byłaby analiza środowiska, w którym układ ma pracować. Należałoby wziąć pod uwagę takie czynniki jak np. materiał, z którego wykonano ściany lub podłoże, aby określić sposób montażu trasy kablowej lub w przypadku elementów będących na zewnątrz budynku, czynniki atmosferyczne przed którym należy chronić urządzenia, jak chociażby zastosowanie hermetycznych puszek w przypadku ryzyka wystąpienia opadów.

4.2 Python

Do implementacji projektu zastosowano wieloparadygmatowy język programowania Python. Powstał on w latach 90-tych ubiegłego stulecia, mając zastąpić język ABC, przede

wszystkim wzbogacając go o rozszerzalność i obsługę wyjątków. Pierwsza wersja oznaczona numerem wersji 0.9 została zaprezentowana przez zespół pod przewodnictwem Guido van Rossuma w roku 1991. Sama nazwa języka wbrew stosowanemu powszechnie logo nie wzięła się od nazwy gatunku węża, lecz od emitowanego od lat 70-tych ubiegłego stulecia serialu pt. „Latający Cyrk Monty Pythona". Wersja 1.0 udostępniona została w 1994, natomiast 2.0 w 2000 roku. Ważną datą dla historii rozwoju tego języka jest rok 2008, kiedy zdecydowano o rozdzieleniu projektu na dwie gałęzie-wersje programu: 2 oraz 3. Oficjalnie wersja 2 przestałą być wspierana w roku 2020, tym samym wersja 3 jest jedyną oficjalnie rozwijaną wersją. Organizacją odpowiedzialną za rozwój języka jest niedochodowa fundacja Python Software Foundation (PSF). Sam proces rozwoju języka prowadzony jest przy zastosowaniu PEP (Python Enhancement Proposal). Jest to dokument zawierający propozycje zmian w języku, poddawany pod dyskusję społeczności programistów, która może zaopiniować wdrożenie lub odrzucenie propozycji. Jednym z najważniejszych dokumentów tego typu jest PEP8, zawierający propozycję organizacji składni, m.in. separację bloków kodu źródłowego za pomocą wcięć oraz kończenie instrukcji nową linią, co czasem powoduje nazywanie Pythona „językiem bez średników", w odróżnieniu od składni popularnych języków programowania wysokiego poziomu jak C# czy Java, w których separacja linii kodu odbywa się za pomocą średnika, a grupowanie poleceń znakiem nawiasu klamrowego „{}" Inną ciekawą cechą odróżniającą Python od wspomnianych tu języków jest dynamiczne typowanie, czyli brak konieczności jawnego zadeklarowanie typu zmiennej, błędnie określanego czasami przez początkujących programistów jako brak typów w Pythonie. W rzeczywistości typ zmiennej przydzielany jest w momencie inicjalizacji zmiennej wartością. Kolejną różnicą jest brak kompilacji, jest to bowiem język interpretowalny. Umożliwia to uruchomienie go w dowolnym środowisku obsługującym interpreter, ale w porównaniu z kompilowaną Javą czyni go nieco wolniejszym. Początkowo określany jako język skryptowy, obecnie przy zachowaniu tej pierwotnej funkcji ma bardzo wiele zastosowań, używany do programowania obiektowego, aplikacji Internetowych, jest także niezwykle popularny w Internecie Rzeczy na platformach sprzętowych Raspberry czy BeagleBoard.

Rysunek 4. 1 Logo języka Python

4.3 Biblioteka GPIO

Jedną z cech, które powodują popularność Pythona na Raspberry Pi jest duża ilość bibliotek wspierających podzespoły powszechnie stosowane w automatyce, robotyce, czy Internecie Rzeczy. Podstawą i punktem wyjścia dla obsługi większości tych urządzeń jest obsługa opisanego już pokrótce interfejsu GPIO. Jest to biblioteka z otwartym kodem źródłowym, łatwo dostępna np. za pomocą managera pakietów *pip3*.

```
pip3 install RPi.GPIO
```

Rysunek 4. 2 Instalacja biblioteki GPIO w systemie Raspbian

Dzięki niej możliwe jest przyporządkowanie fizycznym pinom GPIO numerów wg. jednej z dwóch głównych konwencji, tj. BCM lub BOARD. W sposobie numeracji określanym słowem BOARD piny numerowane są wg. faktycznej kolejności na płytce stykowej, natomiast w domyślnym dla biblioteki standardzie BCM numeracja jest zgodna z ustaloną przez firmę Broadcom, będącą producentem procesora. Mimo wspomnianych tutaj ustawień domyślnych trybu BCM dobrą praktyką jest jawne zadeklarowanie trybu celem uniknięcia pomyłki lub nieporozumienia. Samą numerację w systemie Raspbian można łatwo sprawdzić poleceniem *pinout.*

```
J8:
   3V3  (1) (2)  5V
 GPIO2  (3) (4)  5V
 GPIO3  (5) (6)  GND
 GPIO4  (7) (8)  GPIO14
   GND  (9) (10) GPIO15
GPIO17 (11) (12) GPIO18
GPIO27 (13) (14) GND
GPIO22 (15) (16) GPIO23
   3V3 (17) (18) GPIO24
GPIO10 (19) (20) GND
 GPIO9 (21) (22) GPIO25
GPIO11 (23) (24) GPIO8
   GND (25) (26) GPIO7
 GPIO0 (27) (28) GPIO1
 GPIO5 (29) (30) GND
 GPIO6 (31) (32) GPIO12
GPIO13 (33) (34) GND
GPIO19 (35) (36) GPIO16
GPIO26 (37) (38) GPIO20
   GND (39) (40) GPIO21
```

Rysunek 4. 3 Schemat będący wynikiem zastosowania polecenia *pinout*. Numeracja w konwencji BOARD w nawiasach, na zewnątrz BCM.

```
GPIO.setmode(GPIO.BCM)
```
Rysunek 4. 4 Deklaracja trybu numeracji BCM w projekcie.

Innymi funkcjami oferowanymi przez bibliotekę jest ustawianie pinów jako wejścia lub wyjścia, przypisywanie wejściom stanów (wysoki lub niski), oraz odczytywanie stanów wyjść. Możliwa jest także konfiguracja programowych rezystorów podciągających i ściągających (ang. pul up/down), detekcja zbocza oraz reagowanie na wykryte zbocze.

```
GPIO.setup(DRZWI_PIN, GPIO.IN, pull_up_down = GPIO.PUD_DOWN)
GPIO.setup(ALARM_PIN, GPIO.OUT)
```
Rysunek 4. 5 Przykładowa konfiguracja wejścia oraz wyjścia. Widoczne przypisanie stanu IN oraz OUT pinowi o numerze przypisanym do zmiennej oraz konfiguracja programowego rezystora zwierającego do masy (pull down).

```
GPIO.output(ALARM_PIN, 1)
```
Rysunek 4. 6 Przypisanie stanu wysokiego do pinu wejściowego.

```
while GPIO.input(BUTTON_PIN) == 1:
```
Rysunek 4. 7 Wykorzystanie odczytu wartości pinu wejściowego w pętli while.

```
GPIO.add_event_detect(MQ2_PIN, GPIO.FALLING)
GPIO.add_event_callback(MQ2_PIN, zagazowanie)
```
Rysunek 4. 8 Przykład wykrycia wystąpienia zbocza na zdefiniowanym wejściu oraz reakcja na to zbocze poprzez wywołanie odpowiedniej funkcji.

4.4 Biblioteka I2C_LCD_driver

4.5 Biblioteka adafruit_dht

4.6 Biblioteka datetime

4.7 Biblioteka mfrc522

4.8 Implementacja

Rozdział 5

Testowanie rozwiązania oraz spostrzeżenia

5.1 Wyzwalanie alarmów pojedynczo

5.2 Wyzwalanie więcej niż jednego alarmu jednocześnie

5.3 Kasowanie alarmów

5.4 Odporność na sabotaż

Rozdział 6

Zakończenie

Bibliografia

The Raspberry Pi Foundation. (2021, 05 09). *www.raspberrypi.org*. Pobrano z lokalizacji https://www.raspberrypi.org/blog/8gb-raspberry-pi-4-on-sale-now-at-75/