Instytut Informatyki Politechniki Śląskiej

Zespół Mikroinformatyki
i Teorii Automatów Cyfrowych

Rok akademicki	Rodzaj studiów*: SSI/NSI/NSM	Przedmiot: (Języki Asemblerowe/SMIW)	Grupa	Sekcja
2022/2023	SSI	**SMiW**	**1**	**2**
Prowadzący przedmiot:	Jarosław Paduch		**Termin: dzień tygodnia (godzina)**	

Imię:	**Aleksandra**
Nazwisko:	**Lewandowska**
Email:	aleklew480@student.polsl.pl

Sprawozdanie

Temat ćwiczenia:

„Analiza i wybór elementów”

Skład sekcji:

Aleksandra Lewandowska

TEMAT PROJEKTU:

Celem projektu jest zbudowanie systemu, w ramach którego użytkownik z wykorzystaniem kodu zmiennego jest w stanie otworzyć zamek magnetyczny znajdujących się w drzwiach. Głównym celem projektu jest, aby użytkownik z wykorzystaniem tokenu NFC, przykładając ten token do czytnika współpracującego z mikrokontrolerem był w stanie otworzyć ten zamek. Ważnym elementem systemu jest również zarządzanie metodami dostępu i użytkownikami, aby uprawnieni użytkownicy byli w stanie otworzyć dany zamek.

ANALIZA FUNKCJONALNA:

Aby spełnić wymagania tematu projektu, możemy zdefiniować trzy niezbędne komponenty, z których nasz system powinien się składać.

1. Aplikacja **www/api,** która będzie zawierać niezbędne informacje do otworzenia zamka
2. Układ mikrokontrolerowy, który będzie współpracował z aplikacją www/api w celu zdefiniowania czy podane poświadczenia uprawniają do otworzenia zamka
3. Mechanizm, które umożliwia wyposażenie użytkownika w wymagane poświadczenia, aby użytkownik był w stanie podać odpowiednie poświadczenia (token NFC) i otworzyć zamek

APLIKACJA www/api:

Aplikacja będzie zawierać bazę danych, która będzie przechowywać informację o użytkownikach, o przypisanych uprawnieniach dostępu do odpowiednich zamków (klucze), na podstawie których generowany w czasie rzeczywistym będzie kod TOTP. Zamek oraz token NFC w danej chwili na podstawie posiadanych kluczy powinny być w stanie wyliczyć kod TOTP, i jeżeli obydwa kody są zgodne zamek powinien się otworzyć.
Dodatkowo aplikacja www powinna być wyposażona w możliwość nadawania oraz odbierania uprawnień danemu użytkownikowi.

UKŁAD ZAMKA:

Układ mikrokontrolerowy zamka powinien być wyposażony w czytnik tokenów NFC, na podstawie których będzie odczytywał identyfikator użytkownika oraz kod TOTP. Na podstawie identyfikatora, układ powinien z wykorzystaniem adaptera wifi i dostępu do Internetu wykonać zapytanie do **api.** Na podstawie odpowiedzi serwera w czasie rzeczywistym z wykorzystaniem RTC (real time clock), powinien być w stanie za pomocą otrzymanego klucza wygenerować kod TOTP. W przypadku, w którym obydwa kody się pokrywają, ten dostarczony z wykorzystaniem tokenu NFC oraz wyliczony kod TOTP w czasie rzeczywistym przez układ mikrokontrolerowy powinien spowodować otwarcie zamka magnetycznego oraz zaświecenie się zielonej lampki. W przypadku przeciwnym, użytkownik powinien zostać poinformowany o dostarczeniu błędnych poświadczeń (zaświetlenie się czerwonej lampki).

MECHANIZM ZAOPATRZENIA UŻYTKOWNIKA W POŚWIADCZENIA:

Z wykorzystaniem platformy Android, użytkownik będzie musiał zalogować się do systemu oraz pobrać odpowiednie poświadczenia (klucz), na podstawie którego powinien wyliczyć kod TOTP. Następnie z wykorzystaniem wbudowanego komponentu NFC, powinien

wyemitować informacje o przypisanym identyfikatorze użytkownika oraz wyliczony w czasie rzeczywistym kod TOTP, na podstawie którego układ zamka powinien podjąć decyzję o otwarciu czy zamknięciu zamka.

REALIZACJA

REALIZACJA APLIKACJI WWW/API:
Aplikacja będzie napisana w technologii Asp .Net Core oraz będzie wyhostowana w lokalnej sieci, w której będzie znajdował się mikrokontroler.
Technologia Asp .Net Core jest najnowocześniejszą metodologią pisania aplikacji webowych wspierana przez Microsoft.
Do zalet tej technologii należą:

1. Łatwość w tworzeniu aplikacji, które mogą być hostowane w różnych środowiskach uruchomieniowych (Windows Server, Linux etc.)
2. Technologia jest wspierana w czasie rzeczywistym przez firmę Microsoft
3. Wiele firm korzysta z tej technologii w ramach realizacji wszelakich projektów

Pomimo wielu zalet tej technologii, ważnym komponentem w podjęciu decyzji realizacji aplikacji w tej technologii, jest fakt, że osoba realizująca projekt posiada już doświadczenie w tej technologii.

APLIKACJA DLA UŻYTKOWNIKA KOŃCOWEGO:
Aplikacja będzie napisana na platformę Android, z wykorzystaniem technologii natywnych. Aplikacja będzie napisana w języku programowania Java i zostanie napisana z wykorzystaniem IDE Android Studio. Ta technologia jest wspierana w czasie rzeczywistym przez firmę Google. Dodatkowo warto zwrócić uwagę, że firma Google jest w dużym stopniu odpowiedzialna za tworzenie systemu operacyjnego Android. Oraz fakt, że rozwiązanie będzie implementowane w sposób natywny, daje możliwość w pełni skorzystania z możliwości sprzętowych telefonu, oraz daje szerszy dostęp do API oraz usług znajdujących się w tym systemie operacyjnym.

REALIZACJA UKŁADU MIKROKONTROLEROWEGO ZAMKA

UKŁADY PERYFERYJNE WYJ/WE:
Aby układ spełniał wymagania podane wyżej, musi zawierać funkcjonalność komunikowania się w sieci (łączenie się z api), czytnik NFC (do odczytu tokenów NFC), zegar czasu rzeczywistego (RTC) aby być w stanie w czasie rzeczywistym wyliczyć wartość kodu TOTP. Oraz dodatkową pamięć EPROM do tymczasowego zapisu poświadczeń, w przypadku, gdyby układ został odcięty od serwera.

W ramach oferty serwisu Botland, do dyspozycji mieliśmy:

1. W przypadku zegara czasu rzeczywistego (RTC):
   Większość komponentów zegara czasu rzeczywistego (RTC) korzystają z protokołu I2C jako metodę komunikacji, oraz zapewniają odczyt **sekundy, minuty, godziny, dni, miesiące i lata do roku 2100.** Większość modułów bazuje na RTC DS1307, który jest jednym z bardziej popularnych.
   Przykładem takiego modułu jest:

Grove – DS1307 zegar czasu rzeczywistego I2C

Alternatywnie można skorzystać z bardziej wyrafinowanych modułów zegara czasu rzeczywistego np.

Moduł zegara czasu rzeczywistego RTC RV-8803 Qwiic - SparkFun BOB-16281
Taki moduł zawiera mnóstwo dodatkowych funkcji jak przerwanie odliczania czasu, przerwania alarmu dla dnia tygodnia i etc.

W naszym przypadku wystarczy nam podstawowy moduł DS1307, z faktu, że ten moduł działa tylko na potrzebę wygenerowania kodu TOTP w czasie rzeczywistym. Dodatkowo zaproponowany powyżej moduł należy do cenowo atrakcyjnych tylko **34 PLN,** gdzie droższe bardziej rozbudowane moduły kosztują już **89 PLN**.

2. Dla modułu dostępu do **wifi**
   Przykładowym modułem jest:
   Moduł WiFi ESP8266 + NodeMCU v3 , którego zaletą jest fakt, że do komunikacji wykorzystujemy protokół I2C, dodatkowo sam moduł jest przystosowany do pracy z przykładowym mikrokontrolerem Arduino. Dodatkowo zawiera 10 wyjść GPIO, gdzie każdy może działać jako PWM, I2C czy 1-Wire. Dodatkowo zawiera płytkę NodeMCU, która nie będzie wykorzystywana w tym projekcie. **21.60 PLN**

   Alternatywnie możemy skorzystać z modułu:
   DFRobot FireBeetle ESP32 IOT WiFi, Bluetooth**,** który dodatkowo poza modułem WiFi zawiera moduł Bluetooth, który może znaleźć zastosowanie w innych ciekawy projektach, lub w alternatywnym rozwiązaniu na implementację zamka magnetycznego. Również wspiera protokół I2C. **64.90 PLN**

   Dodatkowo jeszcze możemy skorzystać z modułu:
   ESP32 Thing - moduł WiFi i Bluetooth BLE - kompatybilny z Arduino IDE - SparkFun DEV-13907**,** który funkcjonalnie znacznie przekracza możliwości dwóch modułów wspomnianych wyżej. Zawiera 28 portów GPIO, zawiera tryb uśpienia oraz zawiera sprzętową akceleracją szyfrowania AES, SHA2, ECC i etc. Też zawiera moduły WiFi oraz Bluetooth.
   To będzie moduł, z którego skorzystamy z faktu, że już jesteśmy zaopatrzeni w ten komponent, mimo, że cenowo jest o wiele mniej korzystny niż dwie powyższe propozycje. **149.00 PLN**

3. W przypadku modułu NFC:
   Dobrym i popularnym kandydatem jest moduł:
   Moduł RFID MF RC522 13,56MHz SPI + karta i brelok**,** który zawiera płytkę umożliwiająco zapis i odczyt danych z urządzeń RFID na częstotliwości 13,56 MHz, zasilany jest napięciem 3.3V, dlatego dodatkowo wymaga aby w połączniu z Arduino skorzystać z konwertera napięcia, choć w przypadku Arduino Mega wystawiony jest dodatkowy port z napięciem 3.3V. Z modułem komunikujemy się wykorzystując SPI. Komunikacja ma miejsce do 10 cm. Dodatkowo cały pakiet jest w całkiem przystępnej cenie **14.50 PLN**.

*inne zestawy dość znacznie odbiegają cenowo i nie będą brane pod uwagę w tej analizie

4. Diody
   Zestaw diod LED 5mm - justPi - 30szt.
   4.90 PLN

5. W podstawowej wersji projekt będzie informował użytkownika z wykorzystaniem samych diod. W wersji rozszerzonej należy wziąć jeszcze pod uwagę relay oraz zamek magnetyczny, który zostałby podłączony dodatkowo do mikrokontrolera.

W przypadku analizy prądów i napięć z łatwością możemy skorzystać z konwertera napięcia czy dodatkowego tranzystora, dlatego analiza prądowa nie ma znacznego znaczenia w doborze mikrokontrolera. Ze względu na wymagania urządzeń peryferyjnych potrzebujemy mikrokontroler, który:

1. Jest w stanie obsłużyć komunikację szeregową I2C oraz SPI
2. W przypadku komunikacji I2C jesteśmy w stanie za pomocą dwóch portów obsłużyć kilka urządzeń jednocześnie, w przypadku protokołu RS232 dla każdego urządzenia wymagane byłyby dedykowane dwa porty.
   Do obsługi mamy 3 urządzenia, czyli minimalnie w przypadku I2C wystarczyłyby dwa porty obsługujące Tx, Rx. Maksymalnie potrzebowalibyśmy ich 6, dedykując osobne porty dla osobnych urządzeń.

Patrząc przez to kryterium dobrym kandydatem jest Arduino Mega:
Arduino Mega 2560 Rev3 - A000067**,** który zawiera w sumie 4 czipy obsługujące komunikację seryjną I2C, czy RS232. Dodatkowo zawiera 54 cyfrowe wyjścia/wejścia. Dodatkowo zawiera wyjście 3.3V, co w przypadku niektórych komponentów redukuje potrzebę skorzystania z dodatkowego konwertera napięcia.
W kwestii pamięci:
„Układ Atmega2560 taktowany jest sygnałem o częstotliwości 16 MHz, posiada 256 kB pamięci programu Flash, 4 kB EEPROM oraz 8 kB pamięci operacyjnej SRAM.”
Czyli spełnia nasze wymagania funkcjonalne.

Z faktu, że jesteśmy już zaopatrzeni w ten mikrokontroler, będzie mikrokontrolerem docelowym.

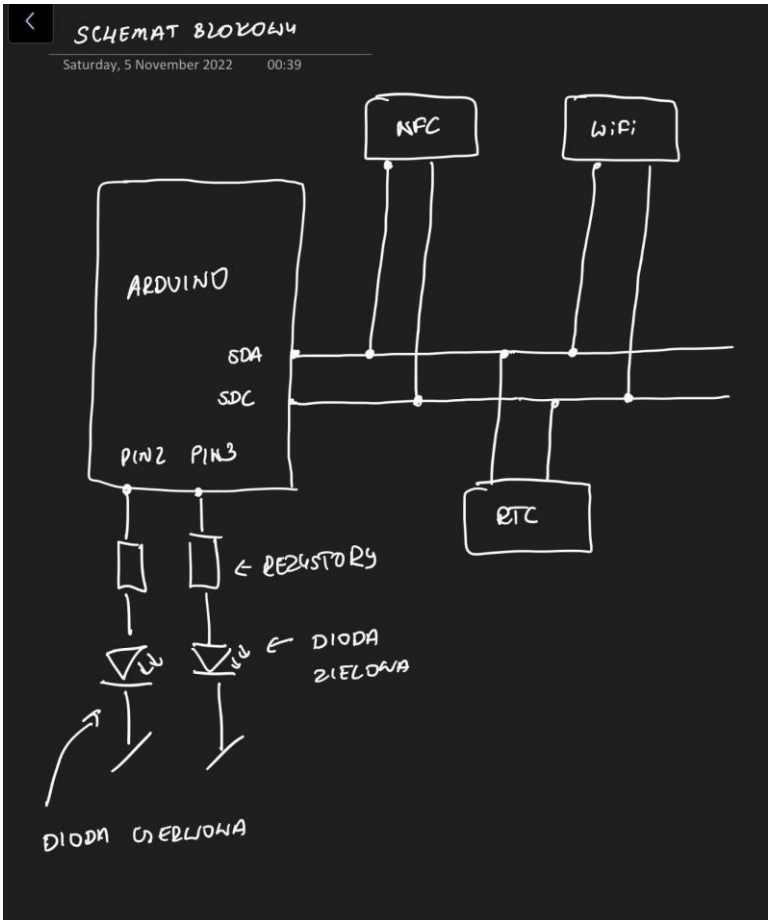
Alternatywnie możemy skorzystać z mikrokontrolera:
Mikrokontroler AVR - ATmega8A-PU DIP
Różnica względem Arduino Mega polega na tym, że mamy do dyspozycji tylko dwa porty komunikacyjne Tx, Rx, chociaż w przypadku obsługi protokołu I2C, też da sobie radę.
„Zasilany napięciem z zakresu od 2,7 V do 5,5 V, taktowany sygnałem do 16 MHz, posiada 8 kB Flash, 1 kB RAM, 512 B EEPROM.”

W porównaniu, Arduino Mega posiada o wiele większy potencjał pamięciowy. Choć cena mikrokontrolera AVR jest o wiele bardziej atrakcyjna **22.20 PLN,** w porównaniu do **249.00 PLN**.

Z faktu, że jesteśmy już zaopatrzeni w Arduino Mega, to będzie mikrokontroler, z którego skorzystamy.

Minimalna cena części do projektu:
34.00 PLN + **149.00 PLN + 14.50 PLN** + **4.90 PLN = 202.40 PLN**
Dodatkowo warto zwrócić uwagę, że nie brane są pod uwagę koszty dodatkowych tranzystorów, kondensatorów, wzmacniaczy operacyjnych czy diod.

SCHEMAT BLOKOWY



Rys. 1 Schemat blokowy układy mikrokontrolera do obsługi zamka

Wszystkie moduły NFC, WiFi, RTC są bezpośrednio połączone z Arduino, do portów obsługi protokołu I2C. Dodatkowo do pinów cyfrowych podłączone są dwie diody świecące, dioda zielona oraz dioda czerwona, które odpowiednio sygnalizują otwarcie oraz zamknięcie zamka.

ZASILANIE

Arduino Mega 2560 Rev3 może być zasilane przez port USB oraz przez zewnętrzne źródło napięcia (non-USB). Płytka operuje w zakresie napięć 6 do 20V, jednakże, kiedy dostarczane napięcie jest mniejsze od 7V, to na wyjściu pinów Arduino możemy otrzymać mniejsze napięcie niż oczekiwane 5V. Jeżeli użyjemy napięcia większego od 12V, regulator napięcia może się przegrzać i uszkodzić płytkę. Dlatego rekomendowany zakres napięć to od 7V do 12V.

Ponieważ wszystkie urządzenia peryferyjne będą zasilane bezpośrednio przez Arduino, dlatego jedynym naszym zmartwieniem jest zasilenie naszego mikrokontrolera.

Jako napięcie stałe dla płytki Arudino wykorzystamy 9V baterię, która znajduje się w oczekiwanym zakresie od 7V do 12V.

Końcówka + baterii powinna zostać podłączona do wejścia V_{in} oraz końcówka – do wejścia GND.