

Laboratorium z przedmiotu Systemy wbudowane (SW)

Karta projektu – zadanie 7

Nazwa projektu:

System alarmowy

Prowadzący:

dr inż. Przemysław
Zakrzewski

Autorzy (tylko nr indeksu):

132247
132211

Grupa dziekańska:

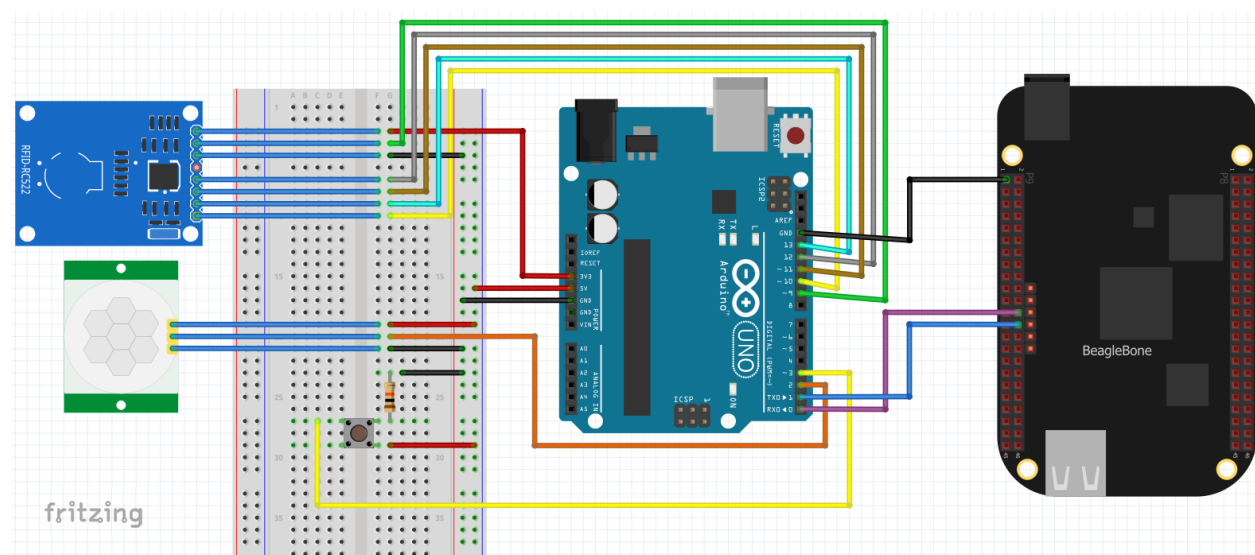
12

Ocena:

Cel projektu:

Celem naszego projektu było stworzenie systemu alarmowego. Do jego stworzenia chcieliśmy wykorzystać czujnik ruchu PIR i czytnik kart RFID do weryfikacji.

Schemat:



Wykorzystana platforma sprzętowa, czujniki pomiarowe, elementy wykonawcze:

Platforma sprzętowa: BeagleBone Black, Arduino Uno

Czujniki pomiarowe: Czujnik ruchu PIR HC – SR501, czytnik kart RFID RC522

Elementy wykonawcze: Przycisk, rezystor 10k Ω , przewody

1. Cel i zakres projektu

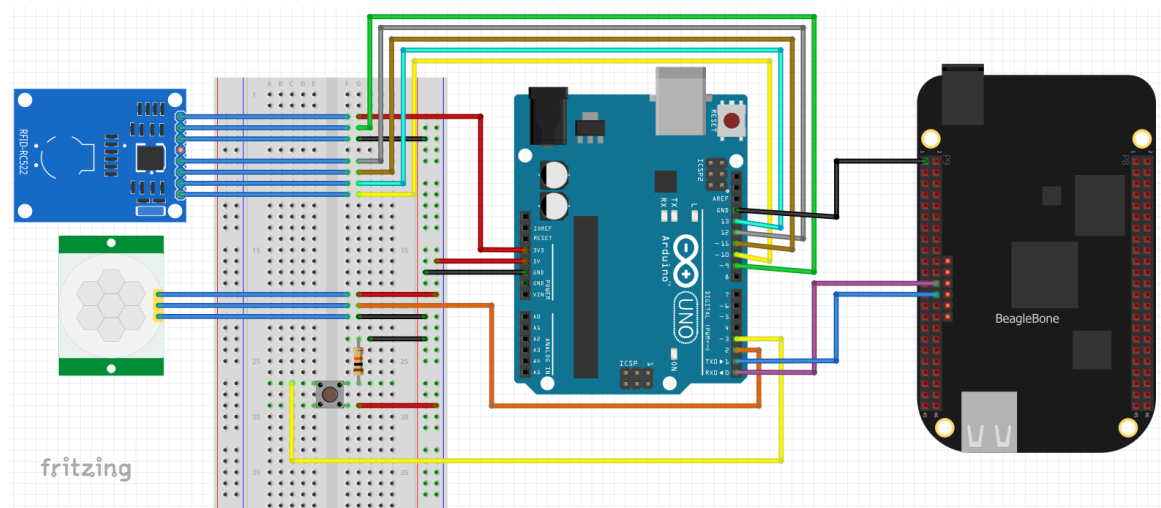
Celem naszego projektu było stworzenie prostego systemu alarmowego.

W jego podstawowej wersji postanowiliśmy skorzystać z dwóch platform sprzętowych BeagleBone Black i Arduino Uno. Celem Arduino było przetwarzanie sygnałów od czujników i przesyłanie tych informacji do BeagleBone'a. Jako czujniki pomiarowe wybraliśmy czujnik ruchu PIR i czytnik kart RFID. BeagleBone Black miał służyć jako "centrum dowodzenia" otrzymując informacje, podejmował on decyzję o dalszym działaniu programu i wysyłał odpowiednie sygnały sterujące do Arduino.

Dodatkowym elementem, który dodaliśmy do naszego układu, był przycisk, który umożliwiał manualne wyłączenie alarmu, gdy nie został jeszcze wykryty ruch.

W dalszej części projektu podjęliśmy decyzję o stworzeniu bazy danych składającej się z dwóch tabel. Jedna zapisuje dane o użytkownikach, jeśli są oni zapisani w słowniku kart, którzy przyłożyli je do czytnika. Jeśli zaś karta nie jest zapisana w tabeli, zostaje zapisany jej numer, a także data. W drugiej została zapisana data i godzina wykrycia ruchu przez czujnik PIR. Postanowiliśmy stworzyć także proste GUI, którego głównym zadaniem jest możliwość włączenia alarmu, sprawdzenie obecnego stanu i możliwość zobaczenia danych z tabel.

2. Schemat



Tablica połączeń:

PIR - Arduino		Przycisk - Arduino		Arduino - BBB	
VCC	5V	VCC	5V	0 (RX)	P9 24(TX)
GND	GND	GND	GND	1 (TX)	P9 26 (RX)
OUT	2	OUT	3	GND	DGND

RFID - Arduino

VCC	3.3V
RST	9
GND	GND
RQ	-
MISO	12
MOSI	11
SCK	13
SDA	10

3. Przejścia stanów

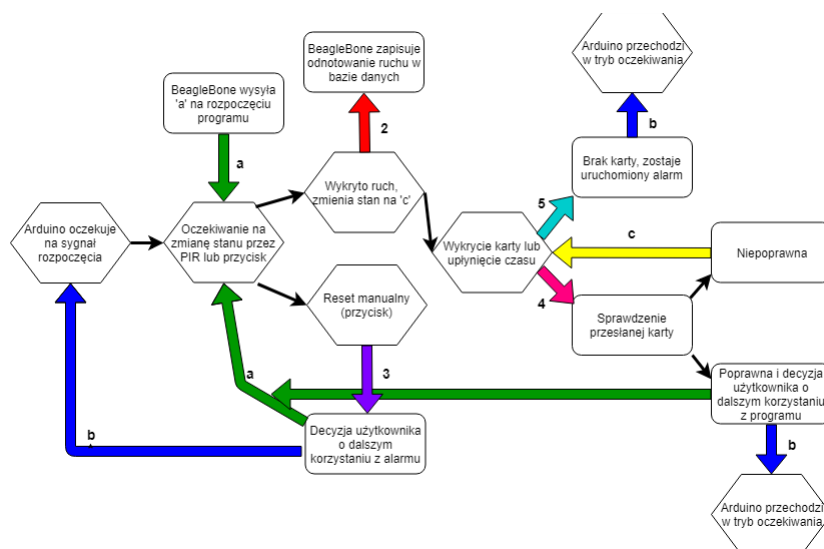
W naszym projekcie stworzyliśmy protokół komunikacyjny.

Arduino:

- 'a' - sygnał oznaczający nasłuchiwanie na przerwanie przez czujnik ruchu i przycisk
- 'b' - sygnał oczekiwania na następne instrukcje. Używany w oczekiwaniu na włączenie alarmu
- 'c' - sygnał oznaczający oczekiwanie na sygnał od RFID

BeagleBone Black:

- 1 – oczekiwanie na sygnał od Arduino
- 2 – wykrycie ruchu. Informacja zostaje zapisana do bazy danych
- 3 – naciśnięcie przycisku. Zapytanie użytkownika o dalszym działaniu programu i wysłaniu odpowiedniej informacji do Arduino ('a' lub 'b')
- 4 – znaleziono kartę. Jeżeli karta jest poprawna następuje zapytanie użytkownika o dalszym działaniu programu i wysłaniu odpowiedniej informacji do Arduino ('a' lub 'b'). Jeśli jest niepoprawna do Arduino jest wysyłane 'c'
- 5 – nie znaleziono żadnej karty. Alarm jest włączony
- 6 – nieoczekiwany błąd



4. Projekt, a realizacja

Ostatecznie udało nam się uzyskać oczekiwany rezultat. Dodaliśmy także nieprzewidziane na początku projektu dodatkowe elementy. Są nimi przycisk do resetu manualnego, baza danych do przechowywania informacji o logowaniach i wykryciu ruchu, a także interfejs użytkownika.

Jednakże osiągnięcie tych celów nie było takie proste, dużym problemem na początku projektu było odpowiednie ustawienie czujnika ruchu, a także ustanowienie odpowiedniej komunikacji pomiędzy platformami przez UART. Reszta podjętych przez nas celów przebiegła względnie bezproblemowo.

5. Ważniejsze fragmenty kodu

Arduino, odczytywanie i wysyłanie informacji:

```
void loop() {  
    Serial.flush();  
    switch (tryb){  
        case WYKRYWAJ_RUCH: {  
            while(tryb==WYKRYWAJ_RUCH){  
                if (digitalRead(PIR) == HIGH) {  
                    Serial.println(RUCH_WYKRYTY);  
                    tryb = 'c';}  
                if (digitalRead(PRZYCISK) == HIGH) {  
                    Serial.println(RESET_PRZYCISK);  
                    odczyt = Serial.read();  
                    tryb = (char)odczyt;}}  
            break;}  
        case ODCZYTAJ_INSTRUKCJE: {  
            odczyt = Serial.read();  
            tryb = (char)odczyt;  
            break;}  
        case ODCZYTAJ_KARTE: {
```

```

time_t start = now();
while((tryb==ODCZYTAJ_KARTE) && (now() - start) <= 30) {
    if ( mfrc522.PICC_IsNewCardPresent()) {
        if ( mfrc522.PICC_ReadCardSerial()) {
            Serial.println(ZNALEZIONO_KARTE);
            Serial.println(mfrc522.uid.uidByte);
            odczyt = Serial.read();
            tryb = (char)odczyt;
            mfrc522.PICC_HaltA();} } }

if(tryb==ODCZYTAJ_KARTE) {
    Serial.println(BRAK_KARTY);
    tryb = ODCZYTAJ_INSTRUKCJE;}
break;}

default: {
    Serial.println(NIEZIDEFINIOWANY_BLAD);
    tryb = ODCZYTAJ_INSTRUKCJE; } } }

```

BeagleBone Black, odczytywanie i wysyłanie:

```
ser = serial.Serial(port="/dev/ttyO1", baudrate=9600)
```

```
ser.write('a')
```

```
while koniec != 1:
```

```
    if tryb == 1:
```

```
        tryb = int(ser.readline())
```

```
    if tryb == 2:
```

```
        tryb = 1
```

```
    if tryb == 3:
```

```
        # Pytanie do użytkownika
```

```
        if(Kontynuuj):
```

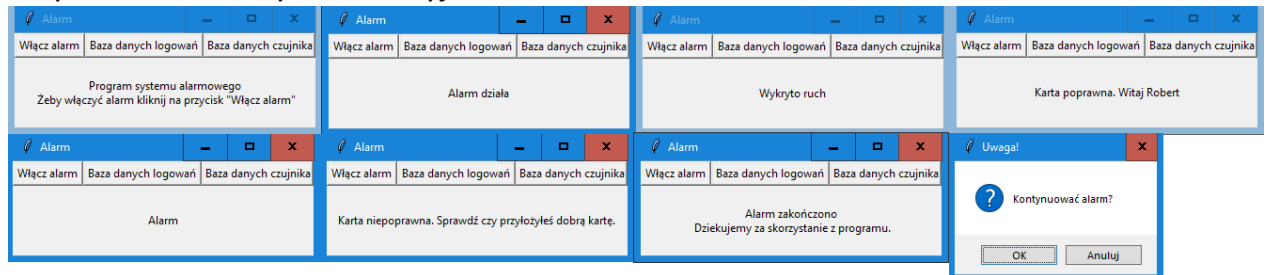
```
            ser.write('a')
```

```
        else:
            koniec = 1
            ser.write('b')
            tryb = 1
    if tryb == 4:
        karta = ser.readline()
        if karta in uzytkownicy:
            user = uzytkownicy[karta]
            # Pytanie do uzytkownika
            if(Kontynuuj):
                ser.write('a')
            else:
                koniec = 1
                ser.write('b')
        else:
            ser.write('c')
        tryb = 1
    if tryb == 5:
        ser.write('b')
        tryb = 1
    if tryb == 6:
        ser.write('b')
        tryb = 1
ser.close()
```

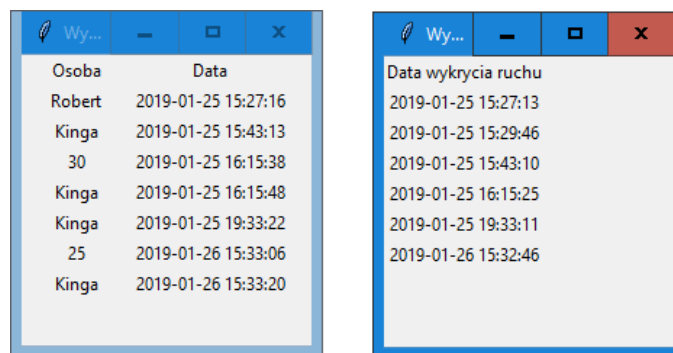
6. Wygląd aplikacji

Aplikacja służy głównie do informowania użytkownika o bieżącym stanie alarmu.

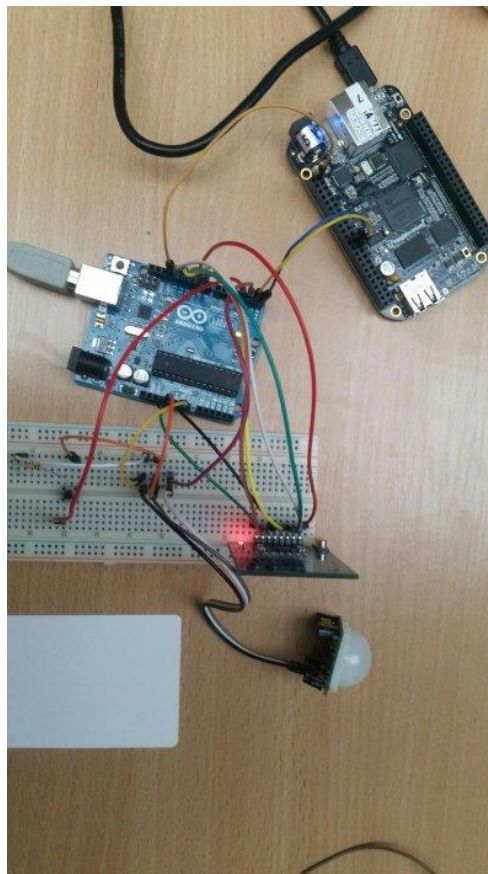
Przykładowe ekrany informacyjne:



Ekran bazy danych logowań i czujnika:



7. Fizyczny wygląd połączeń



8. Podsumowanie

Uważamy, że projekt przebiegł pomyślnie i udało się osiągnąć wszystkie nasze cele. Żeby rozwinąć ten projekt, można byłoby dodać więcej czujników ruchu, a także dodać kamerkę i możliwość nagrywania podczas wykrycia ruchu. Warto byłoby też haszować numery kart, by nie można byłoby się dostać do tej informacji w tak łatwy sposób.

Projekt zainteresował mnie tym tematem i w przyszłości chciałabym zrealizować więcej projektów niekoniecznie związanych z tym konkretnym projektem.