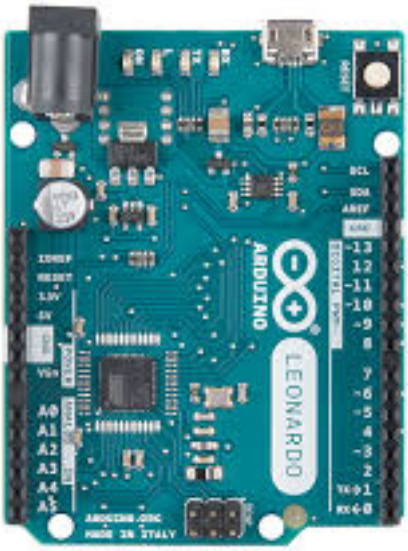
Politechnika Wrocławska



Architektura Komputerów 2

Projekt

Sprawozdanie

Autorzy:
Maciej Białkowski
241285
Dawid Trzebiński
241336

Prowadzący:
dr inż. Dominik Żelazny

Wrocław, 23 maja 2019

Spis treści

1 Założenia projektu

1.1 Opis

Celem naszego projektu było skonstruowanie elektronicznego zamka do drzwi wykorzystującego procesor AVR umożliwiającego autoryzacje osób wykorzystując technologię RFID. Zamek miał być w pełni samowystarczalny - posiadać własne zasilanie oraz przechowywać autoryzowane kody. Po wykryciu znanego ID zamek uruchamia sekwencję napisaną w języku assembly, która uruchamia silnik przesuwający/zamykający zasuwkę.

1.2 Termin I - 01.04.2019

Zbudowanie modelu fizycznego zamka.
Przygotowanie wszelkich komponentów:
- Arduino Leonardo (z mikrokontrolerem AVR Atmega32U4)
- moduł czytnika kart magnetycznych
- bateria
- gniazdo ładowania
- socket na baterię
- stelaż na silnik
- mechanizm zamka

1.3 Termin II - 29.04.2019

Wykorzystanie języka assembly w celu napisania funkcji sterujących silnikiem, brzęczykiem oraz diodami poprzez ustawianie poszczególnych portów jako wyjściowe oraz wprowadzanie ich w stany niskie oraz wysokie.
Wykorzystanie języka C oraz biblioteki MFRC522 obsługującej czujnik kart magnetycznych do napisania ciała programu i wywołań funcji assemblera.

1.4 Termin III - 27.05.2019

Wgranie kodu do mikrokontrolera, wpięcie mostka H pozwalającego na organizację kierunku obrotu silnika. Podłączenie zielonej i czerwonej diody oraz brzęczyka do sygnalizacji otwierania i zamykania zamka.
Sporządzenie sprawozdania

2 Przebieg projektu

2.1 Termin I - 01.04.2019

Udało się zmontować model fizyczny przy użyciu zadeklarowanych komponentów.
Podjęcie decyzji o dodaniu mostka H w III etapie projektu.
Etap oddany w założonym terminie.

2.2 Termin II - 29.04.2019

Udało się napisać założone funkcje sterujące oraz obsługę czytnika RFID.
Etap oddany w załozonym terminie.

FUNKCJE W ASSEMBLY ##############################

```
#include <avr/io.h>

.global pinsSetup
.global delay_500ms
.global delay_1500ms
.global engineStop
.global engineLeft
.global engineRight
.global redLedOn
.global redLedOff
.global greenLedOn
.global greenLedOff
.global buzzerOn
.global buzzerOff


#ustawienie pinow wyjsciowych
pinsSetup:
        sbi DDRD-0x20, 4
  sbi DDRD-0x20, 6
        sbi DDRD-0x20, 7
  sbi DDRB-0x20, 4
  sbi DDRB-0x20, 5
#wywolanie metody stopujacej silnik aby uniknac nieporzadanego uruchomienia
  call engineStop
  ret
```

```
#metody opoznien
delay_500ms:
    ldi  r18, 41
    ldi  r19, 150
    ldi  r20, 128
    call L1
    ret

L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
    ret

delay_1500ms:
    nop
    ldi  r18, 122
    ldi  r19, 193
    ldi  r20, 130
    call L2
    ret

L2: dec  r20
    brne L2
    dec  r19
    brne L2
    dec  r18
    brne L2
    nop
    ret

#zatrzymanie silnika
engineStop:
        cbi PORTD-0x20, 4
        cbi PORTD-0x20, 7
  ret
#uruchomienie silnika w lewo
engineLeft:
        sbi PORTD-0x20, 4
        cbi PORTD-0x20, 7
  ret
#uruchomienie silnika w prawo
engineRight:
        cbi PORTD-0x20, 4
        sbi PORTD-0x20, 7
  ret

#zapalanie i gaszenie diod i brzeczyka
redLedOn:
        sbi PORTB-0x20, 5
  ret

redLedOff:
        cbi PORTB-0x20, 5
  ret
```

```
greenLedOn:
        sbi PORTB-0x20, 4
  ret

greenLedOff:
        cbi PORTB-0x20, 4
  ret


buzzerOn:
        sbi PORTD-0x20, 6
  ret

buzzerOff:
        cbi PORTD-0x20, 6
  ret
```

FUNCKJE W JEZYKU C ##############################

```
#include <SPI.h>
#include <MFRC522.h>
#include "assembler.h"
constexpr uint8_t RST_PIN =  5;          // Configurable, see typical pin layout above
constexpr uint8_t SS_PIN =  10;          // Configurable, see typical pin layout above

MFRC522 mfrc522(SS_PIN, RST_PIN);        // instancja modu u MFRC522
bool state = false; //status zamka(false -otwarty/true -zamkni ty - domy lnie otwarty)
void setup() {
  pinsSetup();        //USTAWIANIE PINOW DO TESTOWANIA AKCJI
        Serial.begin(9600);      // Initialize serial communications with the PC
        //while (!Serial);       //PETLA WYKRYWAJACA BLAD URUCHOMIENIA OKNA DIALOGOWEGO
        SPI.begin();             // Inicjacja port w ICSP(SPI)
        mfrc522.PCD_Init();      // Inicjacja modu u MFRC522
        Serial.println(F("Scan PICC to see UID, SAK, type, and data blocks..."));
        //komunikat okna dialogowego sygnalizujacy poprawne uruchomienie komponentow
}
void loop() {
        // Dodawanie nowej karty
        if ( ! mfrc522.PICC_IsNewCardPresent()) {
                return;
        }
        // wybieranie jednej z kart
        if ( ! mfrc522.PICC_ReadCardSerial()) {
                return;
        }
//operacja po zeskanowaniu karty
if (mfrc522.uid.uidByte[0] == 0x3a &&
     mfrc522.uid.uidByte[1] == 0x98 &&
     mfrc522.uid.uidByte[2] == 0x34 &&
     mfrc522.uid.uidByte[3] == 0x2d) {
     //shardkodowane ID karty kt ra ma dost p
     Serial.println("Card 3A-98-34-2D, access granted");
     if(state){
      Serial.println("Closing...");
      closing();//akcja zamykania zamka
      state = false;
     }
```

```
      else{
       Serial.println("Opening...");
       opening();//akcja otworzenia zamka
       state = true;
      }
}else{
    //niepoprawny adres id wyswietla komunikat i w oknie dialogowym
    //o nieznanym adresie id karty i wyswietla go
      Serial.print("Unknowed card (NR:");
      Serial.print(mfrc522.uid.uidByte[0],HEX);
      Serial.print("-");
      Serial.print(mfrc522.uid.uidByte[1],HEX);
      Serial.print("-");
      Serial.print(mfrc522.uid.uidByte[2],HEX);
      Serial.print("-");
      Serial.print(mfrc522.uid.uidByte[3],HEX);
      Serial.println("), access denied");
      wrongId();
}
}
//niepoprawny adres id, brzeczyk + czerwona doda
void wrongId(){
  redLedOn();
  buzzerOn();
  delay_1500ms();
  redLedOff();
  buzzerOff();
}
//otwieranie i zamykanie zamka + miganie zielon  dioda i brzeczyk
void opening(){
  engineLeft();
  greenLedOn();
  buzzerOn();
  delay_500ms();
  greenLedOff();
  buzzerOff();
  delay_500ms();
  greenLedOn();
  buzzerOn();
  delay_500ms();
  greenLedOff();
  buzzerOff();
  engineStop();
}
void closing(){
  engineRight();
  greenLedOn();
  buzzerOn();
  delay_500ms();
  greenLedOff();
  buzzerOff();
  delay_500ms();
  greenLedOn();
  buzzerOn();
  delay_500ms();
  greenLedOff();
  buzzerOff();
  engineStop();
}
```

2.3 Termin III - 27.05.2019

Udało się połączyć elementy zrealizowane w poprzednich etapach projektu oraz wytrawić mostek H z płytki PCB i podłączyć go do urządzenia.
Napisano sprawozdanie z projektu.

3 Zakończenie projektu

3.1 Podsumowanie

Projekt zrealizowano zgodnie z założeniami i oddano w zadeklarowanych terminach.
Urządzenie działa zgodnie z oczekiwaniami, jest gotowe do prezentacji.

3.2 Napotkane problemy

Powszechnym problemem były niedokładne luty powodujące spadki napięcia oraz luźne elementy konstrukcji. Pomiędzy etapami urządzenie było podatne na uszkodzenia podczas transportu. Przy użyciu niedociętych elementów fizycznych mechanizm zamka działał nieprecyzyjnie i często blokował się - problem znikł po doszlifowaniu i docięciu elementów.