\documentclass[12pt,a4paper]{article}

\usepackage[a4paper]{geometry}

\usepackage[utf8]{inputenc}

\usepackage{lipsum}

\usepackage{listings}

\usepackage{polski}

\usepackage{graphicx}

\graphicspath{ {c:/Users/user/Desktop/Arduino/rfid/DumpInfo/Project-AK2/Lock-Arduino-project/images/} }

\newgeometry{bmargin=2cm, tmargin=2cm, lmargin=2cm, rmargin=2cm}

\begin{document}

\newcommand{\HRule}{\rule{\linewidth}{0.5mm}}

\begin{titlepage}

\center

\textsc{\LARGE Politechnika Wrocławska}\\[1.5cm]

\includegraphics[scale=1]{pobrane}

\\

\vspace{\fill}

\textsc{\Large Architektura Komputerów 2}\\[0.5cm]

\vspace{-3mm}

\textsc{Projekt}

\HRule \\[0.5cm]

{ \huge \bfseries Sprawozdanie}\\[0.2cm]

\HRule \\[1.6cm]

\begin{minipage}{0.4\textwidth}

\begin{flushleft} \large

\emph{Autorzy:}\\

Maciej \textsc{Białkowski} \\ 241285 \\

Dawid \textsc{Trzebiński} \\ 241336 \\

\end{flushleft}

\end{minipage}

~

\begin{minipage}{0.4\textwidth}

\begin{flushright} \large

\emph{Prowadzący:} \\

dr inż. Dominik \textsc{Żelazny}

\end{flushright}

\end{minipage}\\[4cm]

\vspace{\fill}

\textsc{Wrocław, 23 maja 2019}

\end{titlepage}

\tableofcontents

\section{Założenia projektu}

\vspace{-0.7cm}

\makebox[\linewidth]{\rule{\textwidth}{1pt}}

\subsection{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.

\vspace{-0.3cm}

\subsection{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

\vspace{-0.3cm}

\subsection{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.

\newpage

\vspace{-0.3cm}

\subsection{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

\section{Przebieg projektu}

\vspace{-0.7cm}

\makebox[\linewidth]{\rule{\textwidth}{1pt}}

\subsection{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.

\subsection{Termin II - 29.04.2019}

Udało się napisać założone funkcje sterujące oraz obsługę czytnika RFID. \\

Etap oddany w załozonym terminie.

\begin{lstlisting}[basicstyle=\footnotesize]

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

\end{lstlisting}

\newpage

\begin{lstlisting}[basicstyle=\footnotesize]

#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

\end{lstlisting}

\newpage

\begin{lstlisting}[basicstyle=\footnotesize]

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 modułu MFRC522

bool state = false; //status zamka(false -otwarty/true -zamknięty - domyślnie 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 portów ICSP(SPI)

mfrc522.PCD\_Init(); // Inicjacja modułu 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 która ma dostęp

Serial.println("Card 3A-98-34-2D, access granted");

if(state){

Serial.println("Closing...");

closing();//akcja zamykania zamka

state = false;

}

\end{lstlisting}

\newpage

\begin{lstlisting}[basicstyle=\footnotesize]

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 zieloną 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();

}

\end{lstlisting}

\subsection{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.

\section{Zakończenie projektu}

\vspace{-0.7cm}

\makebox[\linewidth]{\rule{\textwidth}{1pt}}

\subsection{Podsumowanie}

Projekt zrealizowano zgodnie z założeniami i oddano w zadeklarowanych terminach.\\

Urządzenie działa zgodnie z oczekiwaniami, jest gotowe do prezentacji.

\subsection{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.

\end{document}