Финальное задание хакатона

**«Короткое замыкание»**

Легенда финального задания хакатона.

В последние годы умные технологии невероятно быстро развиваются и внедряются в разные сферы отечественной экономики. Умные технологии упрощают производственные процессы, помогают повысить эффективность в различных сферах.

**Умный дом** – это идеальное пространство для комфортной среды проживания людей, где технологии работают на их безопасность, экологичность, комфорт.

**1 Задание**

Необходимо собрать и запрограммировать устройство для системы безопасности умного дома. Устройство распознает наличие в нём ключа-карты и открывает запирающий механизм при вводе корректного кода.

Устройство также должно содержать индикаторы, отражающие состояние системы. Если в устройстве отсутствует ключ-карта, то оно должно находиться в полностью выключенном состоянии и не реагировать на действия оператора (индикаторы не горят, механизм закрыт). Если поместить в держатель карточку, должен загореться светодиод наличия карты.

В случае ввода некорректного кода должен загореться светодиод, сообщающий о некорректности введённого кода. Код – последовательность из трёх цифр от 1 до 3, установленная оператора при включении, посредством нажатия на кнопки.

Собранное устройство должно содержать:

1. Слот для ключа-карты, использующий **фотоэлемент** и **светодиод** для проверки наличия ключа-карты в устройстве.
2. **Индикатор 1 (светодиод)**, загорающийся при наличии ключа-карты в слоте.
3. Панель для ввода кода, состоящую из **трёх тактовых кнопок**, пронумерованных цифрами: 1, 2, 3.
4. **Индикатор 2 (светодиод)**, загорающийся только если какая-то из кнопок нажата (индикатор нажатия кнопки). Также, при вводе некорректного кода этот индикатор должен загореться на 3 секунды.
5. **Открывающийся механизм (сервомотор)**, который срабатывает при вводе корректного кода. В открытом состоянии механизм должен находиться 5 секунд после ввода корректного кода, а затем закрываться.

**2 Спецификация необходимого оборудования**

|  |  |
| --- | --- |
| Наименование | Количество |
| Arduino UNO/NANO | 1 |
| Макетная плата | 2 |
| Фотоэлемент (фоторезистор/модуль датчика освещенности) | 1 |
| Сервомотор | 1 |
| Тактовая кнопка | 3 |
| Светодиод | 3 |
| Резистор 150 – 300 Ом | 3 |
| Резистор 1 кОм – 10 кОм | 4 |
| Ключ-карта | 1 |
| Слот для ключа-карты | 1 |
| Ноутбук с предустановленным ПО – Arduino IDE | 1 |

Не допускается использование компонентов не из списка выше.

**3 Разбалловка заданий**

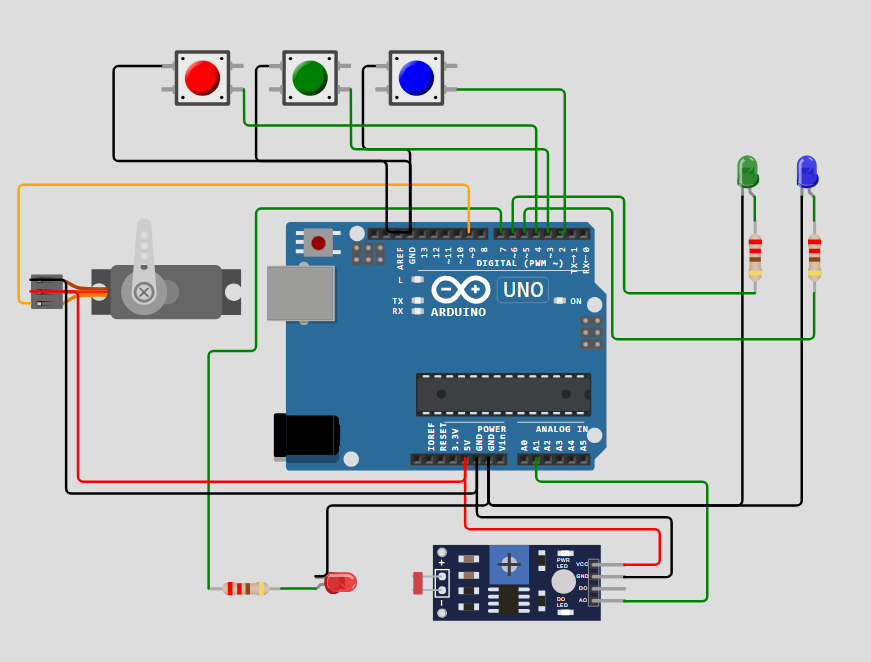
|  |  |  |
| --- | --- | --- |
| Критерии оценки. Максимальное количество баллов **46**  Баллы начисляются за правильный отработанный показ реакции оборудования на запрос судьи или автоматического реагирования оборудования. | | |
| **1 Устройство (10 б)** | | |
| Сборка цепи | 0-8 | Баллы суммируются за каждый пункт:  1 – Верное подключение индикатора ключ-карты.  1 – Верное подключение индикатора состояния системы.  1 – Верное подключение сервомотора.  1 – Верное подключение фотоэлемента.  1 – Верное подключение светодиода для проверки наличия ключа-карты.  1 – За каждое верное подключение тактовой кнопки (макс – 3). |
| Аккуратность проводки | 0-2 | Баллы по каждому пункту НЕ суммируются!  0 – Проводка выполнена неаккуратно, выпадают провода, перемычки не использованы.  1 – Средний уровень. Неоптимальное расположение модулей на макетной плате. Провода закреплены надежно.  2 – Используются перемычки, аккуратно проведена разводка проводов. Монтажные провода не «вываливаются» при перемещении устройства. |
| **2 Программа (6 б)** | | |
| Отладка | 0-3 | Баллы по каждому пункту НЕ суммируются!  0 – в мониторе порта нет отладочной информации.  1 – в мониторе порта присутствует отладочная информация, реализована в меньшей мере.  2 - в мониторе порта присутствует отладочная информация, реализована не в полной мере. Понятно в каком режиме находится устройство  3 – в мониторе порта отладочная информация покрывает все действия оператора. Понятно в каком режиме находится устройство |
| Уровень грамотности кода | 0-3 | Баллы по каждому пункту НЕ суммируются!  0 – начальный. Имена переменных не соответствуют их назначению. Комментариев нет. Не используются константы и массивы. Код программы неуниверсальный.  1 – базовый. Имена переменных логичные. Мало комментариев. Циклы, массивы, функции не используются. Вся логика построена на флагах.  2 – средний. Логика программы построена на флагах. Используются директивы #define и массивы. Действия вынесены в собственные функции. Много комментариев.  3 – высокий. Используется алгоритмы подавления дребезга для обработки кнопки. Логика программы построена на флагах. Обработка событий происходит ассинхронно. Используются собственные функции. Много комментариев |
| **3 Демонстрация функций (24 б)** | | |
| Ключ-карта | 0-4 | Баллы суммируются за каждый пункт:  0 – реакции системы на ключ-карту не наблюдается  1 – устройство не реагирует на действия оператор без ключ-карты  2 – требуемый индикатор 1 загорается при наличии ключ-карты  1 – устройство устанавливает порог освещенности в момент старта. |
| Установка правильного пароля | 0-6 | Баллы суммируются за каждый пункт:  0 – верный пароль не установлен.  1 – первая цифра пароля устанавливается оператором, посредством нажатия на одну из тактовых кнопок.  1 – вторая цифра пароля устанавливается оператором, посредством нажатия на одну из тактовых кнопок.  1 – третья цифра пароля устанавливается оператором, посредством нажатия на одну из тактовых кнопок.  2 – при обработке кнопок присутствует алгоритма debounce (подавления дребезга/зажатия)  1 – после ввода пароля оператором он выводится в монитор порта. |
| Ввод пароля | 0-10 | Баллы суммируются за каждый пункт:  2 - Индикатор 2 (светодиод), загорается только если какая-то из кнопок нажата  2 - При вводе некорректного кода индикатор 2 загорается на 3 секунды  2 – Система успешно фиксирует первую верную цифру введенного пароля  2 – Система успешно фиксирует вторую верную цифру введенного пароля  2 – Система успешно фиксирует третью верную цифру введенного пароля |
| Открывающийся механизм | 0-2 | Баллы суммируются за каждый пункт:  1 – Механизм проходит в открытое состояние после верно введенного пароля  1 – Механизм возвращается в исходное состояние спустя 5 секунд. |
| Цикличность | 0-2 | Баллы суммируются за каждый пункт:  0 – для повторного ввода пароля и открытия механизма нужна перезагрузка платы  2 – после закрытия механизма, устройство автоматически переходи в начальное состояние. Возможно повторное открытия после введенного пароля. |
| **4 Защита работы (6 б)** | | |
| Структурированность защиты | 0-2 | Баллы по каждому пункту НЕ суммируются!  0 – речь не структурирована.  1 – речь структурирована. Объяснены не все этапы работы.  2 – речь структурирована. Объяснены все этапы работы. |
| Обоснованность использования функций и библиотек в коде | 0-2 | Баллы по каждому пункту НЕ суммируются!  0 – использование функции и библиотек никак не аргументировано.  1 – использование функции и библиотек не аргументировано частично.  2 – использование функции и библиотек аргументировано полностью. |
| Аргументированность ответов на вопросы | 0-2 | Баллы по каждому пункту НЕ суммируются!  0 – ответы на вопросы вызывают большое затруднение.  1 – логичность ответов на вопросы. Частичные ответы на вопросы.  2 – уверенные ответы на вопросы. Аргументированность. |

Диск с тестовой программой



**Возможный вариант выполнения работы**

Подключение



**Скетч программы**

/\* Система безопасности с вводом пароля и ключ картой \*/

#include <Servo.h>

#define pos\_1 2 //порт кнопки 1

#define pos\_2 3 //порт кнопки 2

#define pos\_3 4 //порт кнопки 3

#define led\_1 6 //порт индикатора 1

#define led\_card 5 //индикатора карты

#define pinServo 9 //порт сервопривода

#define photo A1 //порт фото

uint8\_t bt1, lastbt1, bt2, lastbt2, bt3, lastbt3; //переменные кнопок

const uint8\_t longpass = 3; //установка длины пароля

int password[longpass]; //массив хранения пароля

uint16\_t brith, porog\_b; //хранение освещенности

uint8\_t queue = 0; //номер вводимой цифры пароля

Servo servo;

void setup() {

Serial.begin(9600);

servo.attach(pinServo);

pinMode(pos\_1, INPUT\_PULLUP);

pinMode(pos\_2, INPUT\_PULLUP);

pinMode(pos\_3, INPUT\_PULLUP);

pinMode(led\_1, OUTPUT);

pinMode(led\_card, OUTPUT);

//Установка уровня освещенности (наличие карты)

Serial.println("Set Brightness, Put card on slot");

delay(3000);

porog\_b = analogRead(photo);

Serial.print("Set Brightness as = ");

Serial.println(porog\_b);

//Установка пароля

Serial.println("Set Password");

set\_password(longpass); //функция вызова настройки пароля

Serial.println("Enter Password");

delay(500);

}

void loop() {

brith = analogRead(photo);

//проверяем наличие карты в слоте

if (brith < (porog\_b + 50)) {

digitalWrite(led\_card, HIGH);

//проверка пароля

bt1 = !digitalRead(pos\_1);

bt2 = !digitalRead(pos\_2);

bt3 = !digitalRead(pos\_3);

if (bt1 || bt2 || bt3) {

digitalWrite(led\_1, HIGH); //включаем индикатор нажатия на кнопку

int cur, flag = 0; //cur - номер нажатой кнопки. флаг - нажатия

if (bt1 && !lastbt1) {

flag = 1;

cur = 1;

}

if (bt2 && !lastbt2) {

flag = 1;

cur = 2;

}

if (bt3 && !lastbt3) {

flag = 1;

cur = 3;

}

//проверяем было ли нажатие

if (flag) {

//проверка совпадения

if (password[queue] == cur) {

queue++; //увеличиваем счетчик правильных нажатий

Serial.println(cur);

}

//неверная кнопка

else {

digitalWrite(led\_1, HIGH);

Serial.println("Wrong");

queue = 0;

Serial.println("Start again");

delay(3000);

digitalWrite(led\_1, LOW);

}

flag = false; //сброс флага, ждем некст нажатия

}

}

else {

// выключаем индикатор если кнопка ненажата

digitalWrite(led\_1, LOW);

}

//если ввели все цифры правильно

if (queue == 3) {

Serial.println("Correct");

queue = 0;

//открыли механизм на 5 сек

servo.write(90);

delay(5000);

servo.write(0);

}

//меняем состояние

lastbt1 = bt1;

lastbt2 = bt2;

lastbt3 = bt3;

}

//все выключаем если карты нет в слоте

else {

Serial.println("OFF");

queue = 0; //сбрасываем счетчик

digitalWrite(led\_1, LOW);

digitalWrite(led\_card, LOW);

}

}

//функция ввода пароля

void set\_password(uint8\_t num) {

//передаем в функцию num - число цифр в пароле

uint8\_t kn1, last\_kn1, kn2, last\_kn2, kn3, last\_kn3, flag;

for (int i = 0; i < num; i++) {

flag = true;

while (flag) {

kn1 = !digitalRead(pos\_1);

kn2 = !digitalRead(pos\_2);

kn3 = !digitalRead(pos\_3);

if (kn1 && !last\_kn1) {

password[i] = 1;

flag = false;

}

if (kn2 && !last\_kn2) {

password[i] = 2;

flag = false;

}

if (kn3 && !last\_kn3) {

password[i] = 3;

flag = false;

}

last\_kn1 = kn1;

last\_kn2 = kn2;

last\_kn3 = kn3;

}

Serial.print("Set");

Serial.print(i);

Serial.print("as = ");

Serial.println(password[i]);

}

Serial.print("Password was set as =");

for (int i = 0; i < num; i++) {

Serial.print(password[i]);

Serial.print(",");

}

}