

Security system

*Охранителна система*

Изготвили:

Анна Донева | Деница Христова | Христо Николов

СЪДЪРЖАНИЕ

* Описание на проекта / Security system…………………………………3
* Съставни части…………………………………………………………..4
* Блокова схема……………………………………………………………5
* Електрическа система………………………………………………...…6
* Сглобяване на охранителната система………………………...………7
* Програмиране на охранителната система /сорс код/..………………...8
* Security system / Заключение…………………………………………..12

ОПИСАНИЕ НА ПРОЕКТА

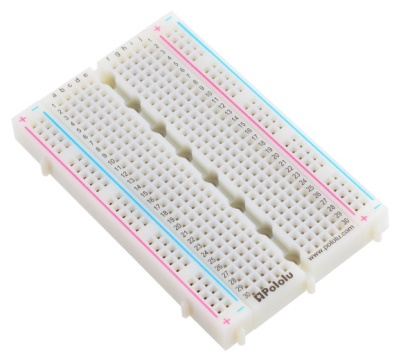
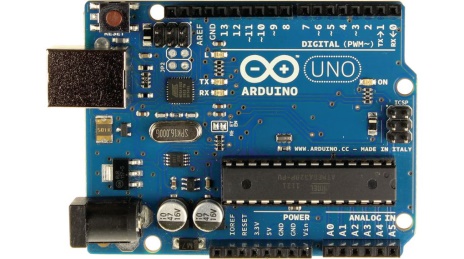
С все по-нарастващата престъпност доверието между хората и чувството за сигурност намаляват. Поради тази причина решихме да създадем охранителна система, която да гарантира сигурността на вас и вашите близки. За целта използвахме Ардуино УНО и няколко сензора, за да конструираме нашата охранителна система. Първото нещо, което си представяме, когато чуем за охранителна система, е досадният и стряскащ звук, който тя издава при засичането на нарушител. Този звук ние постигаме чрез звънец (buzzer), който се активира, ако до определен период от време системата не бъде спряна и изключена чрез карта/чип и бутон. За да е още по-неприятно, но пък сигурно, звънецът издава отново неприятен звук ако картата/чипът за спиране на системата бъде объркан/а. Ако всичко е наред имате достатъчно време да изключите системата чрез бутон и при следващо включване разполагате отново с определено време, преди сензорът за движение да бъде активиран и системата отново задействана.



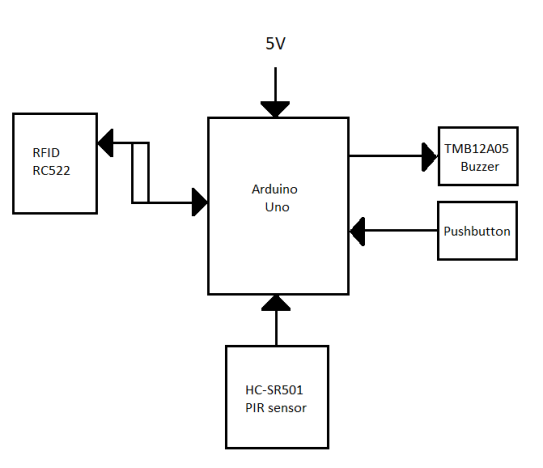
СЪСТАВНИ ЧАСТИ

В изработката на проекта са използвани следните части:

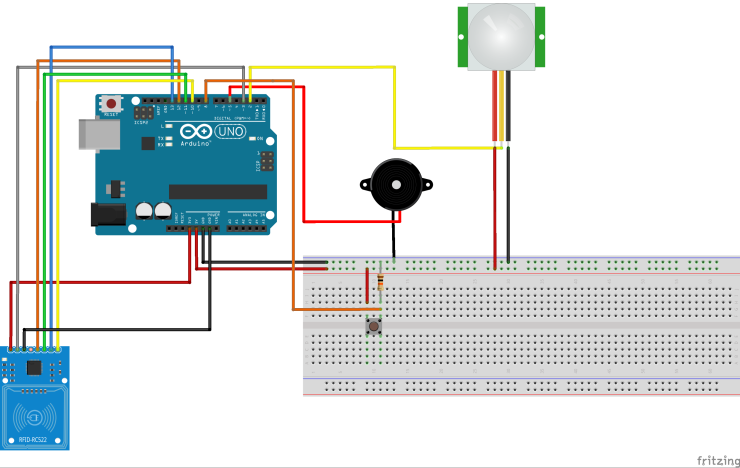
* Arduino Uno (Ардуино УНО платка) – 1бр.
* PIR Motion Sensor (сензор за движение) – 1бр.
* MFRC522 RFID (radio-frequency identification) Reader(радио-честотен идентификатор) 1бр.
* Buzzer (or piezo speaker)(звънец) – 1бр.
* Button (бутон) – 1бр.
* Breadboard (дъска) – 1бр.
* jumper wires (кабелчета)
* 10k ohm resistor (10к ома резистор) – 1бр.
* White Card (карта) – 1бр
* Key Ring(чип) – 1бр.

БЛОКОВА СХЕМА



ЕЛЕКТРИЧЕСКА СХЕМА



СГЛОБЯВАНЕ НА ОХРАНИТЕЛНАТА СИСТЕМА

Breadboard прототипната платка се захранва от Arduino с 5V. За нея е заземен Buzzer, захранващ се от пин D5, и бутон, който получава захранване от breadboard-a, свързан е с пин D8 и заземен чрез 10kOhm резистор.

PIR сензора също се захранва и е заземен в breadboard-a и изпраща сигнала си към пин D2. PIR сензорът е настроен на Non-repeating и при засичане на движение изпраща един продължителен сигнал към микропроцесора по D2.

Описанието на RFID сензора е записано във вид на таблица по-долу. Пиновете нужни за SPI комуникация(MISO, MOSI, SCK и SS) са свързани за дигиталните пинове в адруиното които подържат SPI. RST може да бъде свързан към всеки дигитален пин (трябва да се уточни в кода към кой пин е свързан.)

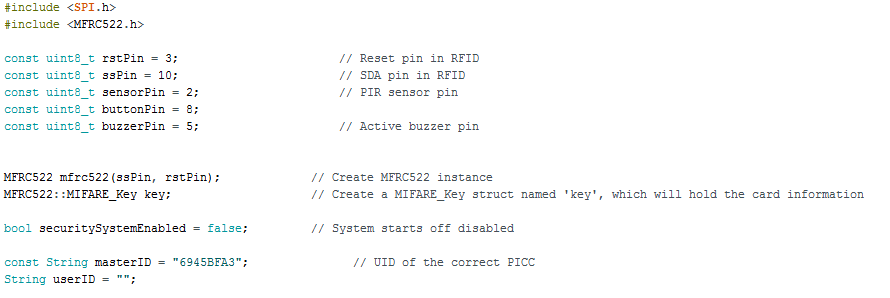
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ПИН | Ардуино  Uno | 3.3V | GND | D3 | Не се използва | D12 | D11 | D13 | D10 |
| RFID-RC522 | 3.3V | GND | RST | IRQ | MISO | MOSI | SCK | SS |

ПРОГРАМИТАНЕ НА ОХРАНИТЕЛНАТА СИСТЕМА/СОРС КОД

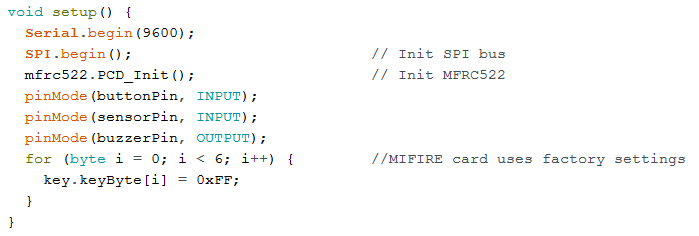
След като бъде включена, системата засича движение от PIR сензора и потребителя има 30 секунди да регистрира присъствието си чрез PICC. Ако той бъде разпознат, потребителя разполага с 30 секунди да изключи системата, за да може тя да не пречи на дейността му в дома/офиса му. В противен случай, ако не се идентифицира с правилния RFID чип, системата за сигурност активира аларма, която уведомява за грешката. Алармата може да бъде прекъсната единствено с правилния RFID чип.

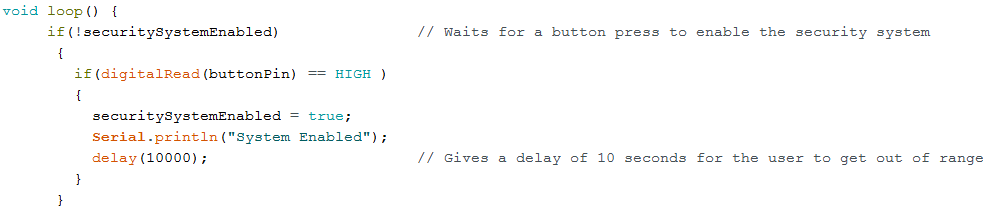
Кодът започва с добавяне на библиотеки за SPI комуникация и за специфичния RFID сензор.

След което се добавят номерата на използваните пинове и се създават обектите, нужни за комуникация с RFID сензора. Създава се променлива, която съхранява информация дали системата е включена, а след това се записва UID на правилния чип.

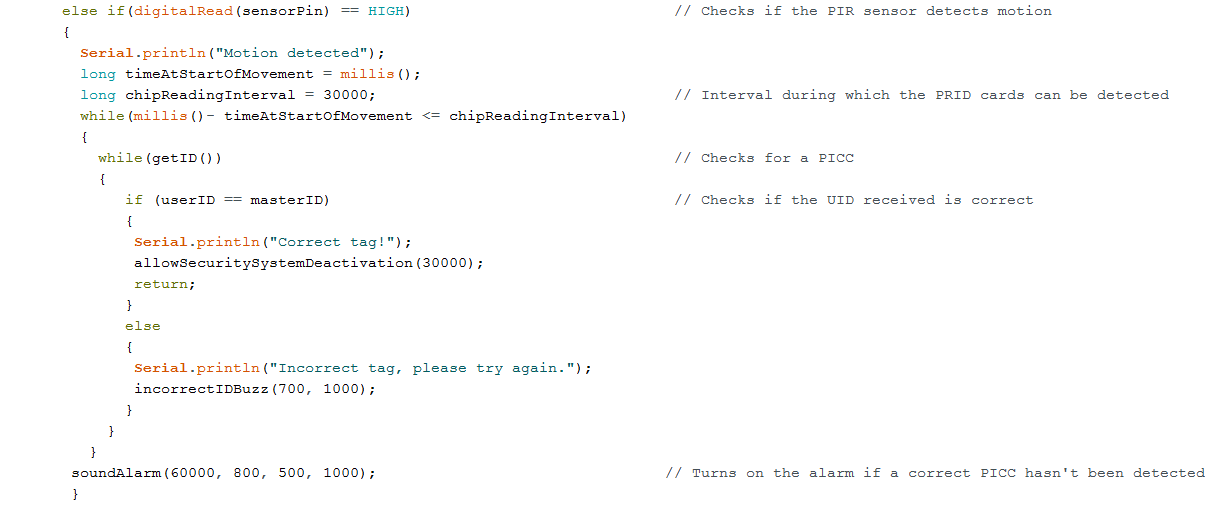
****

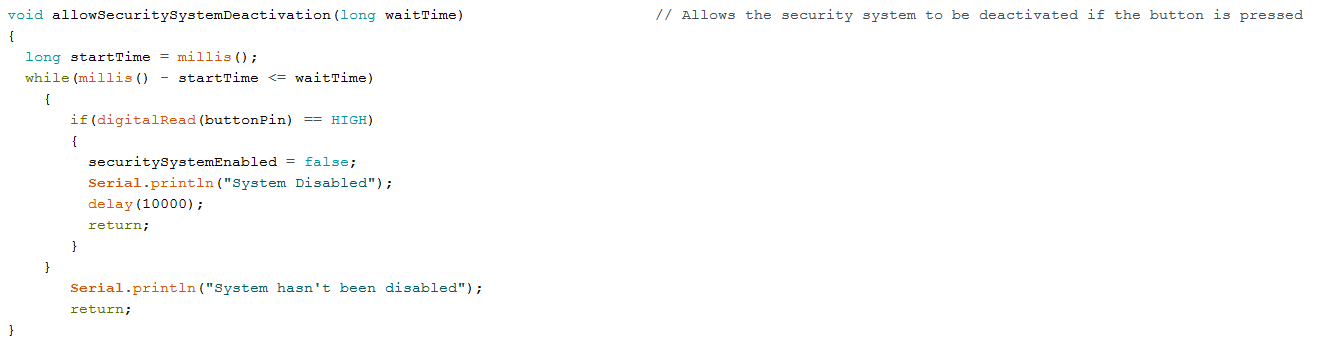
В setup функцията се инициализират обектите, нужни за RFID сензора и серийна комуникация с компюръта. След което задаваме ключа нужен за записване и четене от секторите в нашата PICC. Ние ползваме MIFIRE 1KB карта( фабричните стойности са „FFFFFFFFFF”.)

Loop-ът ни започва с проверка дали системата е активирана и ако не е се проверява за сигнал от бутона. При сигнал се активира системата и се изчакват 10 секунди потребителя да напусне обсега и.

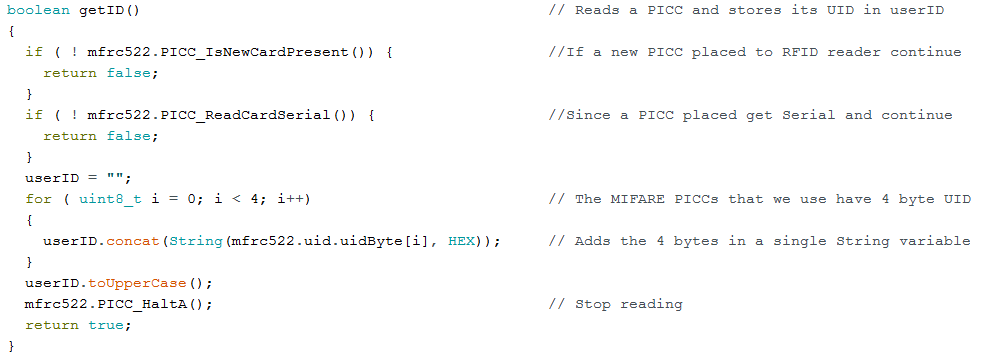


Активираната система за сигурност следи за движение чрез PIR сензора и при засечено движение, започва таймер от 30 секунди, в които потребителят има възможност да подаде своя RFID чин.(изпълнено чрез метода getID()) Ако бъде подаден грешен такъв, buzzer-ът издава предопредителен звук(метод incorrectIDBuzz()). Ако правилният чип бъде подаден, таймерът спира и програмата дава възможност на потребителя да изключи системата(метод allowSecuritySystemDeactivation()). Ако таймерът е изтекъл, алармата ще се активира за 1 минутa(soundAlarm()).

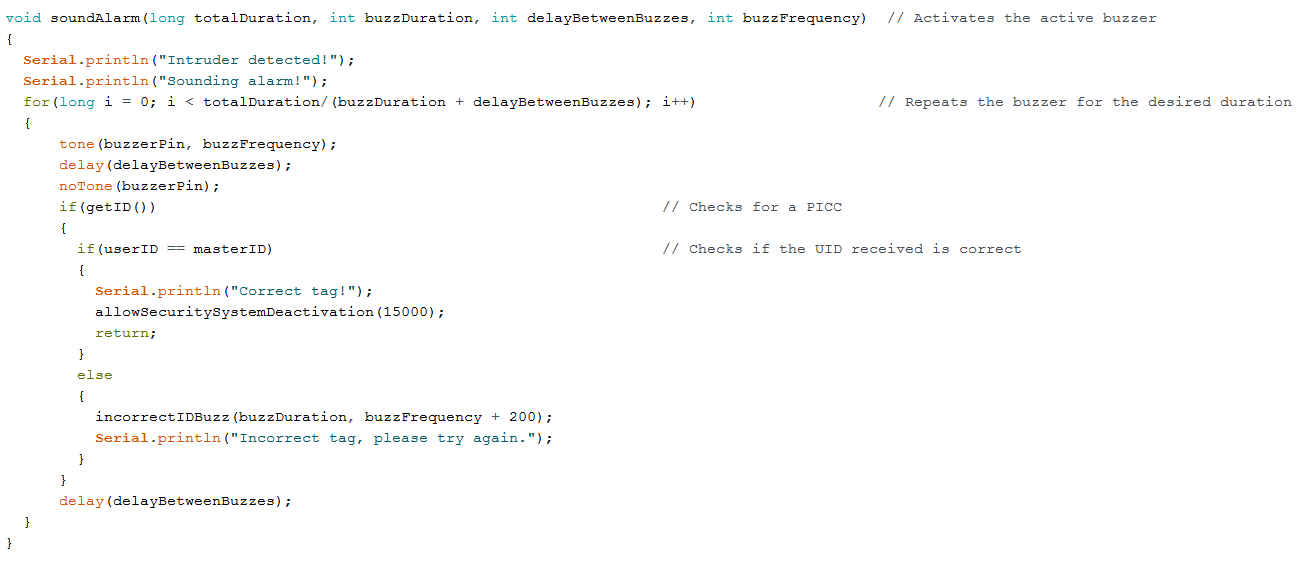


allowSecuritySystemDeactivation() приема един параметър – waitTime. Метода изчаква посоченото време и следи да сигнал от бутона, при който изключва системата. Ако такъв сигнал не бъде подаден системата продължава да следи за движение след изтичане на waitTime.

getID метода проверява за наличието на RFID чип и дали сензора е способен да го прочете. При при неуспех връща false, а ако успее празни променливата, коята съдържа потребителския UID и записва в нея новопрочетения от картата, след което връща true.



Методът, отговарящ за активирането на алармата(soundAlarm()), приема 4 променливи – totalDuration, buzzDuration, delayBetweenBuzzes, buzzFrequency. Фор цикълът се повтаря докато измине totalDuration, като разделяме totalDuration на сбора на buzzDuration и delayBetweenBuzzes. Във всяко изпълнение buzzer-ът издава звук на честотата, определена от buzzFrequency и проверява за RFID чип(getID()). Ако бъде подаден правилния чип, алармата спира и се позволява деактивиране на системата(allowSecuritySystemDeactivation()), а при грешен чип издава звук на малко по-висока честота(incorrectIDBuzz()) за да предупреди потребителя.



incorrectIdBuzz() е прост метод, който ползва buzzer-a да изпълни 1 звук на дадената продължителност и честота.

ЗАКЛЮЧЕНИЕ

Системата успешно засича натрапници и отхвърля карти/чипове с грешна информация, с което успешно минава нашите тестове за сигурност.

