

**Проект по
Вградливи микропроцесорски системи**



Фаза 2: Дизајнирање

*Вграден безбедносен евакуаторен систем со сигурносна движечка
компактна сензорски контролирана тераса во деловен и приватен
простор*

Ментор:

Проф. д-р Моника Симјаноска

Тим:

Зорица Коцева 185043

Андреј Петрушев 181033

Содржина

<i>Архитектура (Architecture)</i>	3
<i>Технички ресурси (Technical Resources)</i>	5
❖ Комплекти за развој (Development Kits)	8
❖ Алатки за развој на софтвер (Software Development Tools)	10
<i>Тим за развој (Development Team)</i>	11





Архитектура (Architecture)

За нашиот проект хардверските и софтверските побарувања како и дополнителни компоненти кои ќе ги користиме при изработка се следниве:





Hardware Requirement

1. Arduino Uno R3
2. Ethernet Shield
3. Bread board
4. LCD 16x2
5. Basic Servo
6. Jumper wire Male to Male 30 to 35
7. Jumper wire Female to Female 20 to 25
8. Jumper wire Female to Male 10-15
9. 4x4 keypad matrix
10. Ultrasonic Sensor - HC-SR04
11. Buzzer piezo
12. LED diodes x2
13. LAN Cable
14. Arduino board power cable
15. RTC ds3231 module
16. Cameras
17. Gas sensor
18. Water sensor
19. Temperature sensor

Софтвер наменет за управување со нашиот систем

1. Arduino IDE -> користиме [IDE Arduino Download page](#)
2. PyCharm IDE -> користиме [IDE PyCharm Download page](#)
*[Micro Python](#) по потреба доколку испрограмираме сензори

Дополнителен материјал за изградба врз кој се прикачуваат и поставуваат хардверските компоненти

- Дрвена куќа макета -> 4x (димензии 22.5 x 30 x 0.2 cm, изградена 18,5x19)



Технички ресурси (Technical Resources)

Proof-of-concept (PoC), исто така познат како доказ за принцип или концепт, е реализација на одреден метод или идеја со цел да се демонстрира неговата изводливост, демонстрација во принцип со цел да се провери дали некој концепт или теорија има практичен потенцијал.

★ Arduino (ќе зборуваме повеќе подолу низ фазата)

★ Cloud Connectivity

Многу „традиционални“ вградени системи биле исклучени системи кои немале пристап до Интернет. Со големиот притисок за IoT, многу системи сега додаваат безжична или жична конекција и проследуваат многу податоци до облакот за обработка и складирање. Традиционалниот вграден развивач на софтвер воопшто нема многу искуство со поставување облачни услуги, работа со MQTT или многу други технологии што се потребни за употреба со облакот, односно:

-Поставување услуги во облак, како што се веб-услуги на Амазон, Google Cloud, итн

-приватни и јавни клучеви заедно со сертификат за уред.

-политика за уреди за уреди што се поврзуваат со услугата облак

-поврзување вграден систем со услугата облак

-пренесување и примање информации од и до облакот

-основна табла за испитување на податоците во облакот и контрола на уредот

★ Security

Со оглед на тоа што многу уреди сега се поврзуваат со облакот, главната грижа со која се соочуваат програмерите е како да ги обезбедат своите системи. Постојат неколку нови технологии, што ќе влијае на тоа како програмерите ги дизајнираат своите системи. Овие технологии варираат од користење на безбедносни процесори, Arm TrustZone и повеќе-јадрени микроконтролери до безбеден и небезбеден код на апликација во партиции. Додека има неколку достапни комплети за хардверска технологија, достапните софтверски решенија се шират со извонредна брзина. Многу од овие технологии штотуку се воведуваат па потребно е да се фокусираме и да ги совладаме безбедносните концепти применувајќи ги вградените системи.



★ Low power design

Вградените дизајнери се занимавале со уреди што работат со батерии, но со повеќе уреди поврзани со IoT и сензорски јазли, дизајнот со мала моќност станал клучен критериум за дизајн со огромно влијание на трошоците на работењето на компанијата. Додека се знае колку малку струја може да извлече микроконтролер во неговиот најдлабок режим на мирување и колку енергетски ефикасни се делови, дизајнирањето на систем што може да ги достигне оние состојби со мала моќност може да биде предизвик.

Програмерите кои работат со уреди што работат со батерии се во неколку клучни области:

- Безжични радио технологии
- Следење на енергијата на хардверот
- Следење на потрошувачката на енергија на софтвер
- Архитектури на батерии
- Регулатори на напојување

★ AI, ML;

Главна тема е движењето на машинското учење од облакот секој дел од системите на модерното време во светот на технологијата. Машинското учење е сила што треба да се смета во облакот и можноста за движење на машинското учење во системи базирани на микроконтролер е менувач.

★ Testing

★ CAN протокол за central locking

Controller Area Network (CAN) е протокол за комуникација со сериски податоци за апликации во реално време. Постојните системи ги проценуваат на сериски магистралаи во однос на нивната можна употреба централно затворање на брави и откриено е дека никој од достапните мрежни протоколи не е во можност да ги исполни барањата. CAN се заснова на „Механизмот за емитување комуникација“, кој се заснова на протокол за пренос ориентиран кон порака. Мрежата на контролорната област (CAN) се користи во широк спектар на вградени, како и системи за контрола на автоматизација.

★ Z-Wave протокол

- ❖ Менаџирање на светла, завеси, порти (со волтажа до 230 V) Менаџмент на air conditioners, underfloor heating systems, heating boilers, radiators, thermostats, итн.



- ❖ Контрола на движење (отворање и затворање на врати/прозорци), temperature, humidity, light intensity, и други сензори

* Како работи Z-Wave протоколот?

- ❖ Z-Wave протоколот се состои од два вида уреди - контролери / master и slave. Контролорот е генерално портал на Z-wave што ја контролира комуникацијата со податоци помеѓу другите јазли и ги поврзува на Интернет. Контролерот доаѓа со претходно програмирана networkID наречена HomeID. Ова е 32-битна идентификација што идентификува одредена мрежа на Z-wave. Slaves се вклучени во мрежата со доделување на HomeID, како и NodeID. NodeID е 8-битен ID кој го идентификува конкретниот јазол или уредот со Z-бран во мрежата. Процесот на вклучување на slave уред во мрежата со доделување на NodeID и признавање на HomeID (специфичен за одредена мрежа на Z-wave) се нарекува „вклучување“. Кога slave-уредот треба да се отстрани од мрежата, HomeID и NodeID се бришат од неа, а уредот е фабрички ресетиран.
- ❖ Уредите со Z-wave можат да комуницираат едни со други само кога имаат ист HomeID. Два уреди со slave Z-wave доделен различни HomeID припаѓаат на различни мрежи на Z-wave и не можат да комуницираат едни со други. Контролорот што доаѓа со одредено пред-програмирано HomeID одржува табела за рутирање за управување со комуникација со податоци помеѓу вклучените јазли, односно уредите со Z-wave вклучени во неговата мрежа. Z-wave користи топологија на мрежна мрежа. Затоа, уредите (slaves) не мора да се поврзуваат со контролорот за да комуницираат со други уреди (slaves) на мрежата. Тие можат директно да комуницираат податоци едни со други. Дури и некои уреди со Z-wave пренесуваат свои сигнали на Z-wave. Таквите уреди се додатоци и дејствуваат како репетитори. Со вакви уреди со Z-wave, контролорот може да комуницира со уредите над неговиот оригинален опсег. Општо, уредите со Z-wave не можат директно да се поврзат на Интернет. Тие се поврзуваат на Интернет само преку контролорот, што е исто така портал на Z-wave кон Интернет.

❖ Комплекти за развој (Development Kits)

Комплетот кој ние ќе го користиме за нашиот проект како развој на системот избравме да биде компатибилен со вежбите кои ги разработуваме, односно го избравме Ардуино како платформа за креирање на проектот.





Со избирањето на Ардуино како позната платформа и достапна за нас студентите за развој имплементација и користење на истата, избрани се соодветни конектори и сензорчиња кои овозможуваат комплетна реализација на она што сакаме да го постигнеме градејќи го системот.

Начинот на кој ќе ги спојуваме самите сензори би бил изведен на неколку начини, односно на breadbord, помеѓу сидовите на макетата да бидат вградени во внатрешниот дел на фасадата и на сидовите, подот или таванот од самата макета.

Arduino комплети за развој на микроконтролер е идеален избор кога сакаме да располагаме со широк спектар на електронски компоненти и да управуваме со интеркомуникацијата на брза, робусна основа. Ардуино има прилично разновидна линија на микроконтролери, почнувајќи од екстремно малечки (Ардуино микро) до неверојатно моќни (Ардуино Дуо). За разлика од Raspberry Pi, плочите Arduino не работат со оперативен систем и доаѓаат со ограничен простор за меморија RAM / Flash (обично во килобајти).

Уникатен фактор: Arduino е робусна, флексибилна и достапна платформа за развој, поддржана широко од растечката мрежна заедница. Бидејќи го користат милиони ентузијаста широм светот, многу е веројатно дека ќе најдете изворен код за случајот за употреба на Интернет, и само треба малку да го измените за да одговараат на вашите барања.

Одговара за: Почетници, кои бараат да го изградат својот прв проект за електроника

Препорачан модел (стартер): Arduino Uno

Поддржани програмски јазици: C / C ++

Arduino Uno R3 како табли за развој на:

- Одбор за развој што е едноставен, достапен и лесен за употреба
- Контрола на мал број влезно-излезни уреди (аналогни / дигитални) и сакате систем што ефикасно управува со интеркомуникацијата.
- Да се создаде проект што не е поврзан на Интернет и работи самостојно на самостојна основа.
- Комплет за развој без дополнителни трошоци за оперативниот систем.

❖ Алатки за развој на софтвер (Software Development Tools)

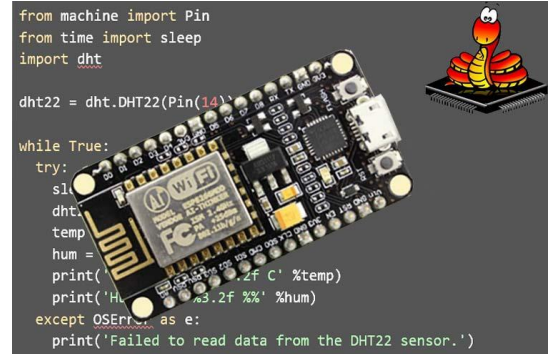
За развој на нашиот вграден систем ќе користиме C, C++ и по можност на некои сензори кои треба да се кодираат би користеле и Python, односно Micro Python.

За развојот на софтвер ќе ги искористиме следните алатки:

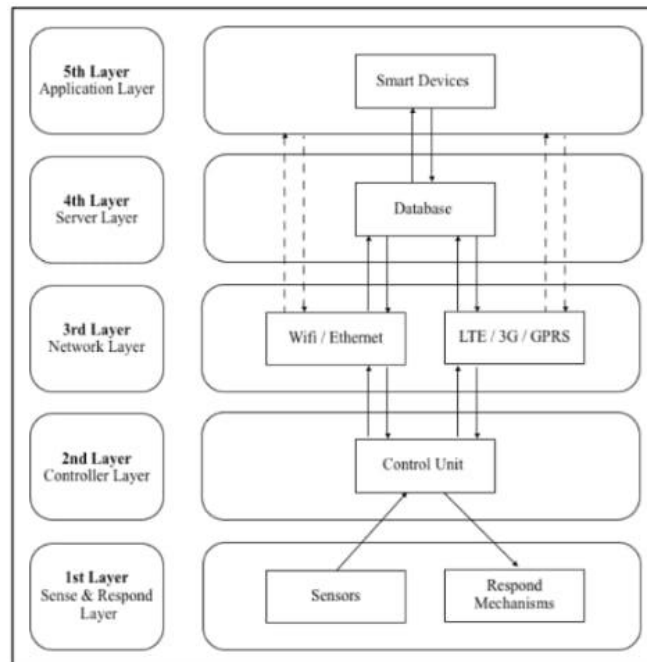
Како оперативен систем би користеле RTOS односно оперативен систем во реално време за извршување на сите функции и задачи што е можно понавремено и поточно.

Во делот на програмски јазици ќе користиме C, C++, како и Python за кодирање на системот/колото.

Како развојни околина или IDEs ќе користиме PyCharm, Arduino итн.



Слоевит хардверски дијаграм





Тим за развој (Development Team)

Да се состави и изгради успешен вграден тим за развој на софтвер за компанија која нема искуство со развој на вграден софтвер е исклучително предизвикувачка задача. Притоа се вклучени многу ризици поврзани со сигурноста и безбедноста на вградените уреди. Спротивно на тоа пак, компаниите за ИТ authorising со широко портфолио на вградени проекти, развиваат голем број најдобри практики. Тие даваат сигурност дека испорачаните решенија одговараат на постојните стандарди за безбедност и индустрија.

Што не прави добри за вградени софтверски инженери?

Кога станува збор за ангажирање тим за развој на софтвер за вграден проект, треба да се биде специфичен за технолошки барања кои задолжително одредени вештини да ги има искусен софтверски инженер за вградени системи:

- 3-5 годишно искуство со програмирање C / C ++;
- Познавање на основни протоколи: I2C, SPI, UART, LIN и протоколи: SATA, PCIe, USB, CAN, MOST;
- Искуство со архитектури на микропроцесор и микроконтролер;
- Експертиза во оперативни системи со општа намена (GPOS) и оперативни системи во реално време;
- Познавање на мрежни технологии како што се Етернет, TCP / IP;
- Солидно разбирање за различните видови на меморија, вклучувајќи RAM, ROM и Flash;
- Познавање на модели на дизајн и алгоритми;
- Основно и солидно познавање на други програмски јазици кои можат да бидат комплементарни, како што се Java, .Net или Python.

Конкретно за нас, ние сме тим од двајца студенти кои преку факултетот ја имаме можноста да направиме и развиеме вграден сигурносен систем кој ќе овозможи пред се, зголемување на безбедноста во рамки на приватните и деловните простории. Исто така, со понатамошен развој на ваков систем ќе се придонесе за многу подобро справување со природните непогоди, како и брза ефикасна евакуација со што ќе се избегнат било какви опасности по човечкиот живот.

Како тим за развој на овој систем, имаме големи познавања од областа на софтверско и хардверско инженерство кои ќе ни овозможат многу полесна изработка и составување на потребните делови за самиот систем. Дополнително, имаме и докажана способност во дизајнот на системот, фирмверот. Нашата комбинирана сила во развојот на вграден софтвер и услугите за дизајн и развој на хардвер нè прави идеални партнери за развој на

IoT производ и вграден или R&D софтвер во реално време. Како инженерски тим можеме да го покриеме целиот животен циклус на производот, од концепт и прототипови до развој, тестирање, производство и поддршка.

Сервисите на вградените системи бараат експертиза која што е различна, но и поврзана во одредени области:

РАЗВОЈ НА СОФТВЕР

- Инженерство на системот
- Софтвер за контрола во реално време
- Инженеринг на фирмвер
- Возачи на уреди и BSP (пакети за поддршка на табли)
- Протоколи за комуникација

РАЗВОЈ НА ХАРДВЕР

- Индустриска електроника и компактни уреди
- Телеметрија / биотелеметрија и телемеханика
- Ултра-нисконапонски уреди
- Груба електроника
- Брзо прототипирање

