

**Гимназија „Јован Јовановић Змај“**

Нови Сад

**Матурски рад из Примене рачунара  
Интегрисани систем за озвучење установе**

Професор ментор:  
Вукашин Деканић

Ученик:  
Филип Параг IV-9

Нови Сад, мај 2018. год.

## ПРЕДГОВОР

При крају другог разреда, одлучио сам да се прихватим прављења система за школско звоно, јер је апарат намењен тој сврси био изразито непрактичан: није се могао мењати распоред звоњења, а промена тренутне мелодије звона је значила долазак сервисера и вишенедељно чекање. Почетно решење сам у школу донео сутрадан, али је оно било далеко од савшеног; није се могло контролисати без мукотрпне системске администрације и није имало режим радне суботе.

Уз помоћ професора и ментора Вукашина Деканића, овај пројекат је временом сазревао, до тренутка када смо одлучили да га излажем као матурски рад. Посебну захвалност желим да изразим целом особљу и свим ученицима Гимназије који су били несвесно учествовали у развијању и побољшавању овог система.

## САДРЖАЈ

	Страна
ПРЕДГОВОР.....	2
САДРЖАЈ.....	3
Увод.....	4
1. Технологије и појмови.....	5
1.1. Разбери Пај.....	5
1.2. Разбијан ГНУ/Линукс.....	7
1.3. Пајтон.....	8
2. Инсталација и подешавање.....	9
2.1. Преузимање системске слике.....	9
2.2. Припрема меморијске картице.....	10
2.3. Повећавање фајлсистема.....	10
2.4. Подешавање система и повезивање на мрежу.....	11
2.5. Инсталација софтвера.....	11
3. Опис имплементације софтвера.....	14
3.1. School bell.....	14
3.2. Config.....	15
3.3. Schedule.....	16
3.4. Ringer.....	17
3.5. Raspberry.....	18
4. Хардверска управљачка плоча.....	21
4.1. Тастери.....	21
4.2. Развојна плоча.....	22
4.3. Повезивање са микрорачунаром.....	23
4.4. Упутство за употребу.....	24
Закључак.....	25
Садржај ЦД-а.....	26
Литература.....	27
Слике.....	27
БИОГРАФИЈА МАТУРАНТА.....	28

## **Увод**

Тренутно се у већини школа користи стандардно електрично звоно за означавање почетка и краја часова. Док је овај начин веома поуздан, недостаје му могућност репродуковања мелодије, односно произвољних звукова на већ постојећој мрежи звучника за разглас.

Овај рад представља интегрисан систем за оглашавање и контролу мелодичног звона имплементиран у програмском језику Пајтон на микрорачунару, који се тренутно користи у Гимназији "Јовај Јовановић Змај" у Новом Саду.

У првом делу разраде описане су употребљене технологије и дефинисани су појмови потребни за разумевање рада. Ту спадају коришћени програмски језик и оперативни систем.

Други део описује ток инсталације и подешавања микрорачунара за непрекидан вишемесечни самосталан рад.

Софтверска имплементација система и дискусија појединачних функција је описана у трећем делу разраде.

Четврти део се бави хардверском управљачком плочом и њеним повезивањем са софтверским делом система.

## 1. Технологије и појмови

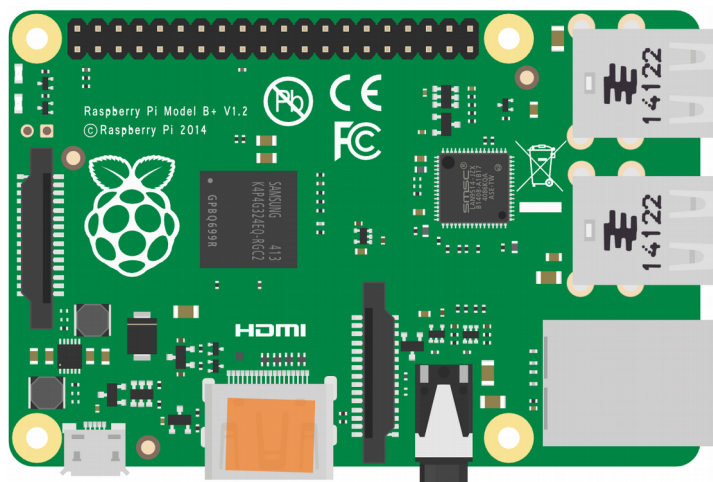
### 1.1. Разбери Пај

Идеја о малом и приступачном рачунару, јавила се 2006. године, када су Ибен Аптон, Роб Мјулинс, Џек Ленг и Алан Мајкрофт са Универзитета у Кембриџу постали забринути нивоом предзнања студената који су се пријављивали за рачунарске науке.

За разлику од 1990. године, када је већина кандидата имала солидно предзнање из области програмирања, 2000. године је просечан кандидат имао мало знања о програмирању. Закључили су да се формирала армија информатичара са врло мало практичног програмерског знања потребног за исправно решавање конкретних изазова. Они су имали искуства у коришћењу кућног рачунара или конзоле за игру и забаву. Искуство стечено свакодневним корисничким радом на рачунару давало им је лажну слику о личном знању из области рачунарских наука.

Претходне генерације училе су програмирање на легендарним Спектрумима, Комодорима 64 или Амига рачунарима, који су осим могућности забаве обезбеђивали и добро окружење за писање првих програма. Због тога су дошли на идеју да направе нешто што је ученицима приступачно по цени, а пружа окружење спремно за писање програма. Од 2006. до 2008, дизајнирано је неколико верзија од којих је настао рачунар Разбери Пај.

Када су се 2008. године појавили процесори који су могли да се користе у мобилним уређајима, приступачни по цени и довољно снажни да обезбеде добру подршку графичком окружењу, постало је извесно да ће пројекат заживети. Када се на тржишту појавио први примерак рачунара Разбери Пај, својим изгледом и могућностима привукао је велику пажњу, како у области обуке програмера, тако и код ентузијаста. Пошто је сам уређај изазвао велику популарност, на тржишту постоји много пратеће опреме, а на интернету је објављено мноштво конкретних савета и практичних упутстава.



Слика 1 - Илустрација микрорачунара Разбери Пај 3

Модел 3, који је објављен у фебруару 2016. године има разноврстан избор прикључака: 40 општенаменских пинова, *RJ45* мрежни прикључак, четири *USB 2.0*, *HDMI*, 3.5-милиметарски звучни *AUX* прикључак, нестандартни прикључак за екран и камеру. Поред жичног повезивања, Пај 3 има и антене за Блутут и бежичну мрежу. У табели су приложене детаљне спецификације микрорачунара.

Табела 1 - Хардверске спецификације модела 3

Назив	Особине
SoC	Broadcom BCM2837
CPU	4 x 1.2 GHz ARM Cortex A53, ARMv8
GPU	Broadcom VideoCore IV, 400 MHz
RAM	1 GB LPDDR2-900 SDRAM
USB	4 x 2.0
GPIO	40
Ethernet	10/100 Mb/s
WiFi	1x1 802.11n
Audio	HDMI, 3.5 mm AUX
Video	HDMI, composite
Bluetooth	4.0

## 1.2. Разбијан ГНУ/Линукс

Постоји више оперативних система прилагођених микрорачунару Разбери Пај. За почетнике је најбољи избор *NOOBS (New Out Of the Box Software)*, који је поједностављен начин за одабир система. За напредније кориснике, на располагању су оперативни системи: Разбијан, Убунту, Виндовс Кор, *OSMC*, ЛибреЕлек, Пинет, РискОС, Федора и други. На слици испод је приказано графичко корисничко окружење новог издања Разбијана.



Слика 2 - Графички кориснички интерфејс Разбијана

### 1.3. Пајтон

Пајтон је веома популаран интерпретирани програмски језик високог нивоа опште намене направљен од стране Гида фон Росума 1991. године.

Пајтон 3 подржава више стилова програмирања: императивни, процедурални, објектно-оријентисани и функционални. Филозофија синтаксе Пајтона подразумева читљивост кода и могућност реализације веома комплексних идеја у малом броју линија кода.

```
def dfs(graph, start):
    visited, stack = set(), [start]
    while stack:
        vertex = stack.pop()
        if vertex not in visited:
            visited.add(vertex)
            stack.extend(graph[vertex] - visited)
    return visited
```

#### Код 1 - Рекурзивна дубинска претрага графа

Програмски језик Пајтон се пласирао међу најкоришћеније програмске језике у образовању на универзитетском нивоу у Сједињеним Државама. Ово се посебно односи на почетне курсеве програмирања. Пајтон није чисто образовни програмски језик. Лакоћа којом се учи и користи, као и његова велика практична примена, довели су до тога да Пајтон преузме вођство од Јаве на курсевима учења програмирања.

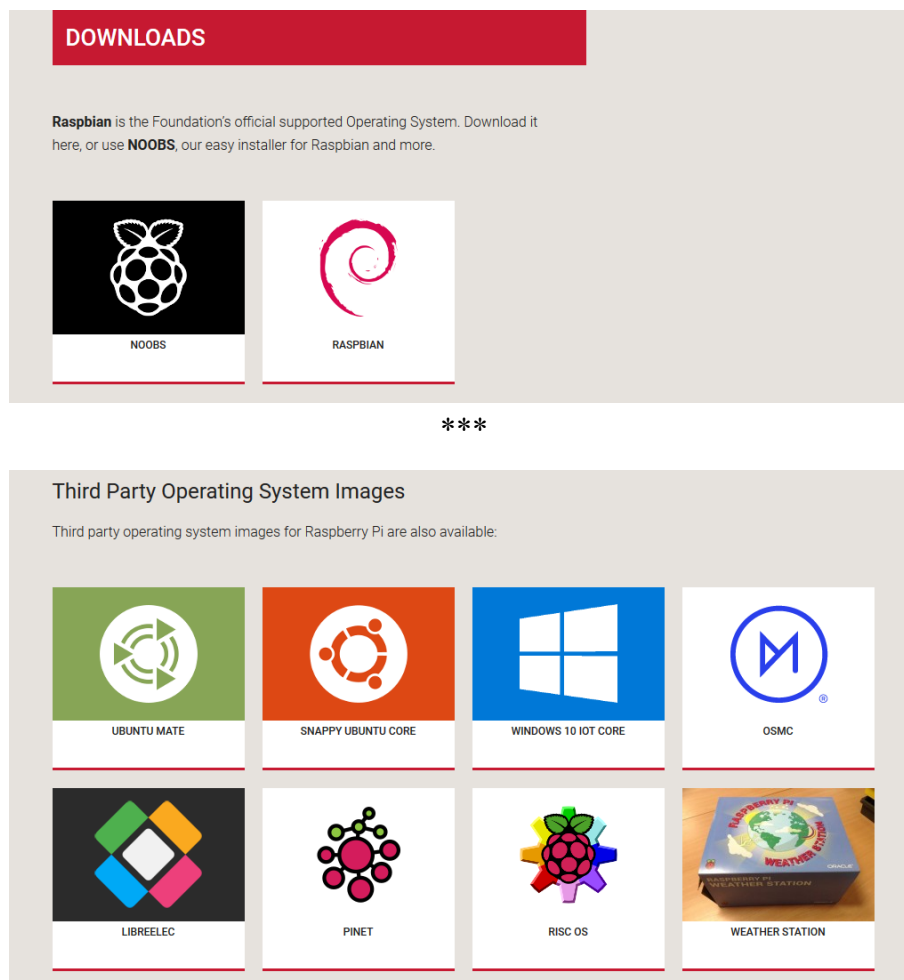
Од 2003. године, Пајтон се непрекидно налази међу десет најпопуларнијих програмских језика на ТИОБЕ листи. Велике организације као што су Гугл, Церн и Наса користе Пајтон у многе сврхе, укључујући и истраживање.



## 2. Инсталација и подешавање

### 2.1. Преузимање системске слике

За разлику од традиционалних персоналних рачунара, паметни телефони и микрорачунари немају управљачки софтвер *BIOS* који се користи за промену системских подешавања, учитавање и одабир оперативног система. Због тога се са веб странице Фондације Разбери Пај могу преузети потпуне системске слике намењене за копирање на меморијску картицу.

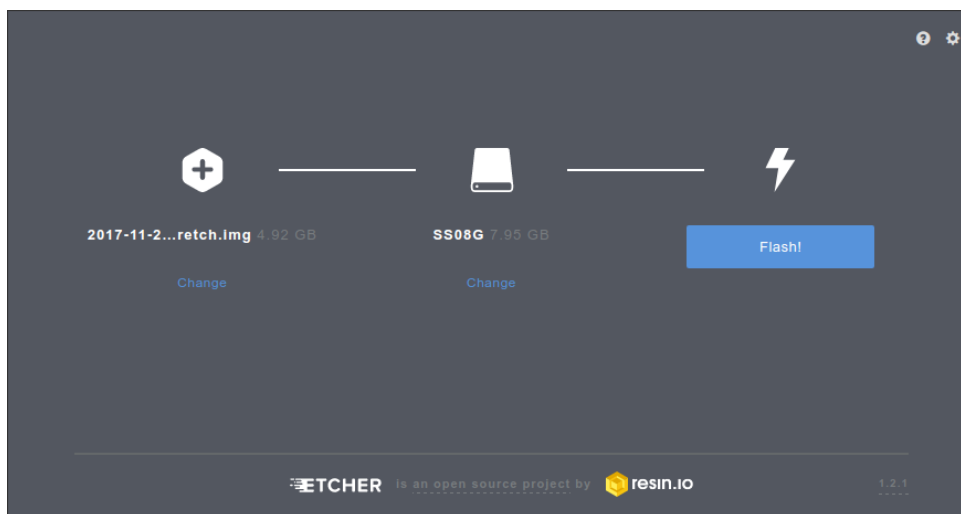


Слика 3 - Веб страница за преузимање системских слика

Као што је наведено раније, постоје више званично подржаних оперативних система. У овом раду се користи Разбијан Стреч, односно девето издање Дебиан дистрибуције прилагођено Разбери Пају.

## 2.2. Припрема меморијске картице

Копирање системске слике на меморијску картицу је извршено помоћу програма Ечер, који пружа веома једноставан и брз начин да се картица форматира и да се на њу копира садржај изабране слике.

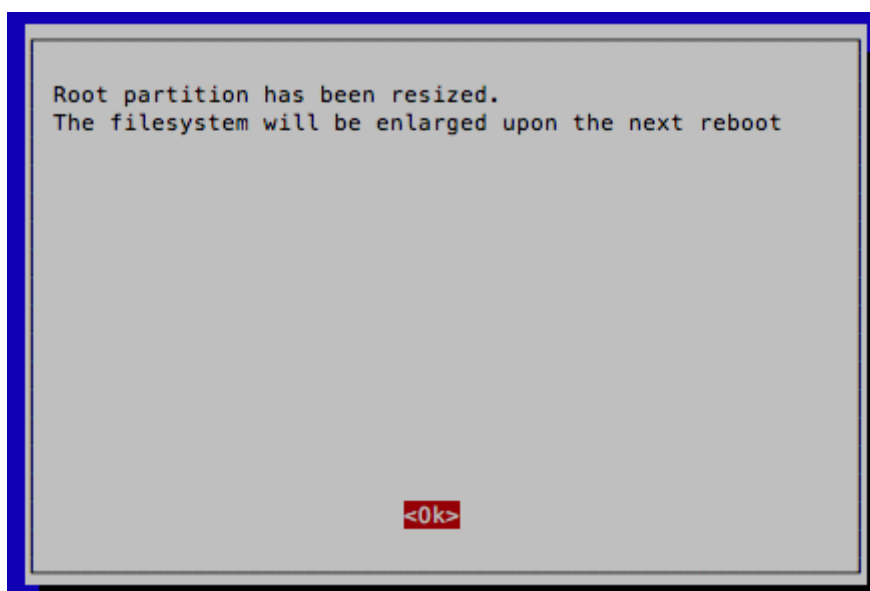


Слика 4 - Копирање системске слике програмом Ечер

Процес копирања траје око двадесет минута због велике системске слике и недовољно брзе картице. Након копирања, овај програм аутоматски проверава да ли су преписани подаци исправни.

## 2.3. Повећавање фајлсистема

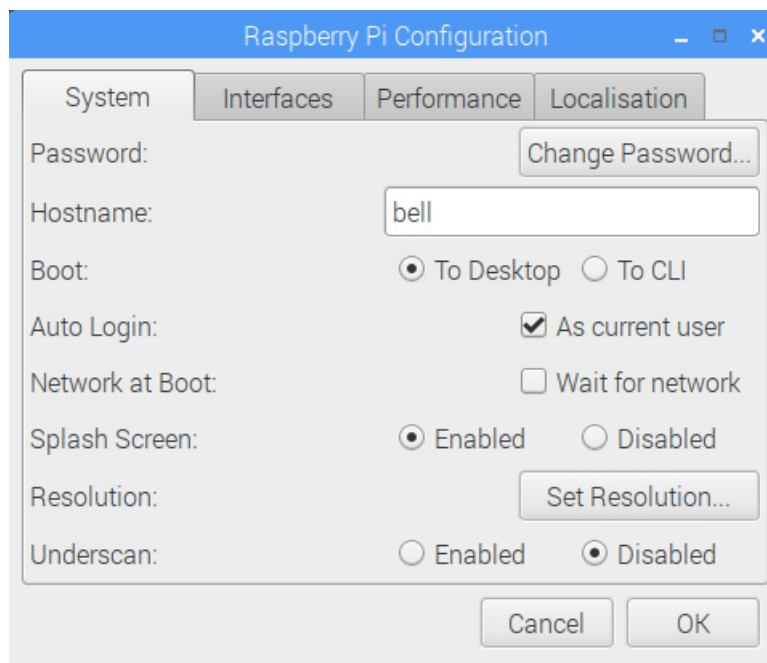
Када се Пај први пут укључи са новим системом, оперативни систем аутоматски проширује фајлсистем тако да заузима сав слободан простор. Уколико је проширење било успешно, појавиће се порука приказана на слици.



Слика 5 - Обавештење при првом укључивању Разбијана

## 2.4. Подешавање система и повезивање на мрежу

Следећи пут када се систем укључи, по подразумеваним подешавањима ће учитати кориснички графички интерфејс. На графичком панелу за подешавања *Raspberry Pi Configuration* су промењени: име микрорачунара, распоред тастатуре и подразумевана резолуција.



Слика 6 - Графички панел за подешавање система

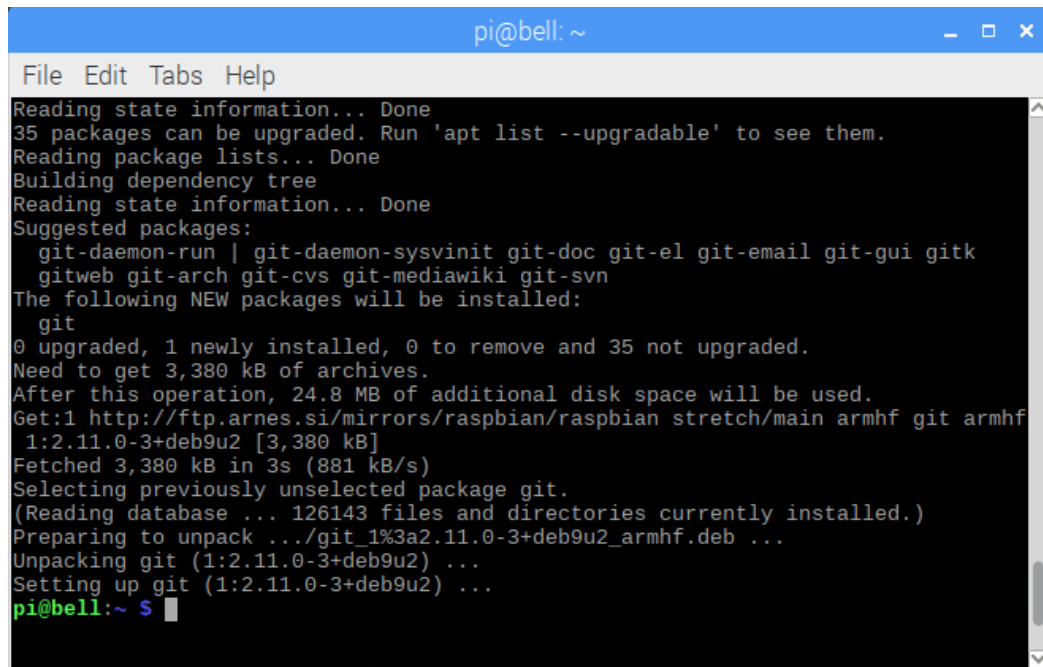
Након тога је микрорачунар повезан на интернет преко бежичне мреже. Стална веза са интернетом је потребна како би системски сат био синхронизован, али и како би се могла вршити дијагностика без физичког приступа систему.

## 2.5. Инсталација софтвера

Софтвер написан у овом раду је јавно доступан на Гитхабу, а за преузимање најновије верзије и за могућност даљег ажурирања, потребно је на систем инсталирати софтвер за контролу верзија кода Гит. На Разбијану се ово најлакше ради извршавањем следеће команде у системском терминалу:

```
sudo apt update && sudo apt -y install git
```

Испис ове команде у терминалу је приказан на следећој слици.



```
pi@bell: ~  
File Edit Tabs Help  
Reading state information... Done  
35 packages can be upgraded. Run 'apt list --upgradable' to see them.  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Suggested packages:  
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk  
  gitweb git-arch git-cvs git-mediawiki git-svn  
The following NEW packages will be installed:  
  git  
0 upgraded, 1 newly installed, 0 to remove and 35 not upgraded.  
Need to get 3,380 kB of archives.  
After this operation, 24.8 MB of additional disk space will be used.  
Get:1 http://ftp.arnes.si/mirrors/raspbian/raspbian stretch/main armhf git armhf  
  1:2.11.0-3+deb9u2 [3,380 kB]  
Fetched 3,380 kB in 3s (881 kB/s)  
Selecting previously unselected package git.  
(Reading database ... 126143 files and directories currently installed.)  
Preparing to unpack ../git_1%3a2.11.0-3+deb9u2_armhf.deb ...  
Unpacking git (1:2.11.0-3+deb9u2) ...  
Setting up git (1:2.11.0-3+deb9u2) ...  
pi@bell:~$
```

Слика 7 - Испис команде за инсталацију програма Гит

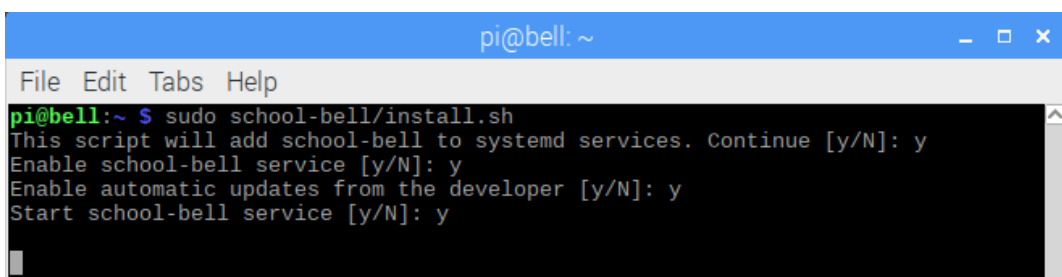
Након што је Гит успешно инсталиран, репозиторијум софтвера описаног у раду се може преузети извршавањем команде:

```
git clone https://github.com/filiparag/school-bell.git
```

Уз софтвер је написана и скрипта за инсталацију, која по жељи корисника може додати софтвер у листу програма покренутих са системом. Инсталација се покреће командом:

```
sudo school-bell/install.sh
```

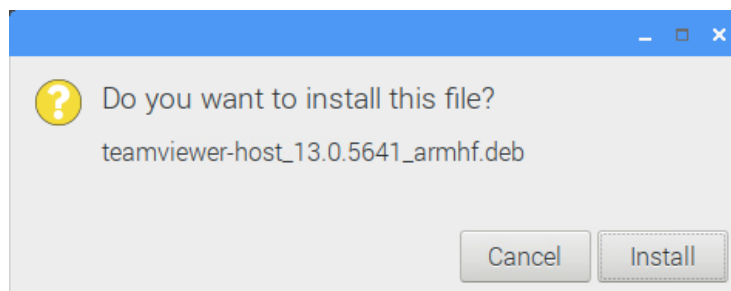
На следећој слици је приказан исход инсталације када је потврдно одговорено на сва питања, односно опције.



```
pi@bell: ~  
File Edit Tabs Help  
pi@bell:~$ sudo school-bell/install.sh  
This script will add school-bell to systemd services. Continue [y/N]: y  
Enable school-bell service [y/N]: y  
Enable automatic updates from the developer [y/N]: y  
Start school-bell service [y/N]: y
```

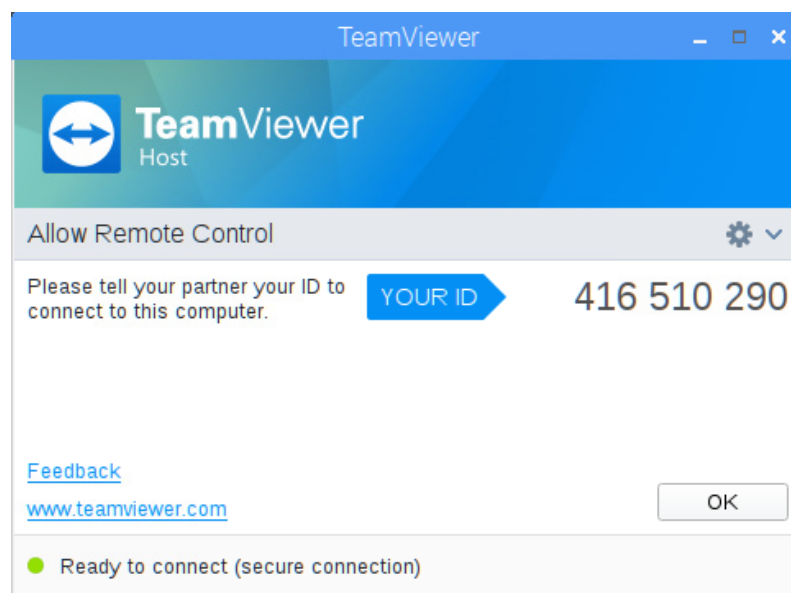
Слика 8 - Инсталација софтвера

За надгледање и евентуалну дијагностику система се могу користити разни софтверски алати, али је због једноставности употребе коришћен Тимвјувер. Преузета инсталациона датотека се дуплим кликом покреће, а затим је потребно потврдити инсталацију.



Слика 9 - Упитни прозор за потврду инсталације Тимвјувера

За приступање микрорачунару је сада довољно запамтити идентификациони број и постављену лозинку. Подешавање лозинке и забрана приступа се могу пронаћи у прозору Тимвјувера.



Слика 10 – Почетни прозор Тимвјувера

### 3. Опис имплементације софтвера

У наредних шест подналова је описан код по целинама, односно по начину на који је раздвојен у датотеке. За сваку функцију је дато објашњење сврхе, аргументи које она очекује и евентуална повратна вредност.

#### 3.1. School bell

##### **Main**

##### Опис функције:

При покретању, програм започиње извршавање ове функције. Из ње се затим позивају све друге функције у току рада: прво се учитавају подешавања и распоред звоњења, а затим се у бесконачној петљи сваки дан по задатом распореду укључује звоно. Уколико је систем у режиму рада на микрорачунару, промена тренутног стања је могућа помоћу хардверске управљачке плоче.

Аргументи: нема

Повратна вредност: нема

## 3.2. Config

### Get

#### Опис функције:

Уколико подешавања нису учитана, учитава их и затим враћа вредност везану за дати кључ који се налази у одређеном одељку подешавања.

Аргументи: одељак, кључ

Повратна вредност: вредност везана за кључ

### Set

#### Опис функције:

Уколико подешавања нису учитана, учитава их и затим у учитаним подешавањима мења вредност везану за задати кључ. Ради мање вероватноће уношења грешке у систем, промењена подешавања се чувају привремено у радној меморији.

Аргументи: одељак, кључ, вредност

Повратна вредност: нема

### Verbose

#### Опис функције:

Уколико је у подешавањима омогућен испис у конзолу или у записни дневник, ова функција задату поруку прослеђује у стандардни излаз, односно терминал или у датотеку записа. Поруке су намењене за дијагностику и проверу исправности софтвера.

Аргументи: порука

Повратна вредност: нема

### Load

#### Опис функције:

Позивом функције за учитавање, из датотеке се подешавања учитавају у радну меморију, након чега су приступна остатку функција уз помоћ претходно описаних функција *Get* и *Set*.

Аргументи: нема

Повратна вредност: нема

### 3.3. Schedule

#### Today

##### Опис функције:

Ова функција враћа распоред звоњења за тренутни дан у виду реда сортираног по времену. Како систем дозвољава да свако звоно има другачију мелодију, поред времена, за свако звоно се додаје и име мелодијске датотеке из директоријума *Sounds*.

Аргументи: име распореда

Повратна вредност: распоред звоњења и задатих мелодија

#### List names

##### Опис функције:

Позивањем ове функције се добија листа имена свих распореда учитаних у радну меморију. Уколико распореди нису учитани, позива се функција за њихово учитавање.

Аргументи: нема

Повратна вредност: листа имена свих распореда

#### Load

##### Опис функције:

Слично *Load* функцији у целини за подешавања, ова функција из свих датотека у директоријуму *Schedules* учитава распореде звоњења. Хронолошко уношење имена није потребно јер се приликом учитавања она сортирају.

Аргументи: нема

Повратна вредност: нема

#### Wait tomorrow

##### Опис функције:

Уколико је у делу програма потребно сачекати крај дана, употребљава се ова функција. Садржи петљу која се извршава све док се не промени тренутни дан или док се не активира зауставни сигнал.

Аргументи: зауставни сигнал

Повратна вредност: нема



### 3.4. Ringer

#### Start

##### Опис функције:

Главна функција за извршавање звоњења, покреће се из *Main* функције сваки дан. Распоред звоњења се редом извршава, све док се ред не испразни. За проверавање времена користи функцију *Current time*, а за репродукцију мелодије функцију *Ring*.

Аргументи: распоред звоњења, зауставни сигнал

Повратна вредност: нема

#### Ring

##### Опис функције:

У зависности од одабраног начина репродукције у подешавањима, ова функција позива спољни програм за репродукцију задате мелодије.

Аргументи: мелодијска датотека

Повратна вредност: нема

#### Current time

##### Опис функције:

Функција као повратну вредност враћа уређен пар тренутног времена у формату сат, минут, секунд. У подешавањима је могуће на тренутно време додати позитиван или негативан временски офсет ради синхронизације са школским сатовима.

Аргументи: нема

Повратна вредност: тренутно време

### 3.5. Raspberry

#### **GPIO setup**

##### Опис функције:

Уколико је у подешавањима укључен режим рада на микрорачунару Разбери Пај, потребно је на почетку конфигурисати општенаменске пинове да би били у конфигурацији за касније описану хардверску управљачку плочу. Након подешавања пинова, у зависности од подешавања, микрорачунар се поставља у подразумевано или претходно стање.

Аргументи: нема

Повратна вредност: нема

#### **GPIO cycle**

##### Опис функције:

Постоје више начина за интеракцију са хардвером, ондосно општенаменским пиновима. Један од њих је да се све радње извршавају периодично, односно да се промене не дешавају у реалном времену по деловању, већ у тачно одређеним интервалима. Ова функција служи за позивање других функција за интеракцију у одређеном реду уколико је режим рада активан.

Аргументи: режим рада

Повратна вредност: нема

#### **GPIO connected**

##### Опис функције:

Провера да ли је хардверска управљачка плоча повезана на микрорачунар.

Аргументи: нема

Повратна вредност: истинитосна вредност повезаности

#### **GPIO poll**

##### Опис функције:

Ова функција служи за читавање вредности тастера и бележење стања.

Аргументи: нема

Повратна вредност: нема

#### **GPIO button state**

##### Опис функције:

Функција враћа вредности тастера добијених у претходно описаној функцији као низ.

Аргументи: нема

Повратна вредност: низ вредности тастера

### **GPIO led**

Опис функције:

Обрнуто функцији за читавање вредности, ова функција поставља вредности пинова светлећих диода на задате.

Аргументи: нема

Повратна вредност: нема

### **GPIO button power**

Опис функције:

Мења вредност режима рада са активног на искључен и обрнуто. Као и наредне функције, позива се сваки пут када је притиснут респективни тастер, у овом случају за режим рада.

Аргументи: нема

Повратна вредност: нема

### **GPIO button schedule**

Опис функције:

Поставља тренутно активан распоред на следећи по абecedном реду. Позива се сваки пут када је притиснут тастер за режим рада. Уколико је активна радна субота, промена се врши над распоредом за суботу, у супротном за све дане.

Аргументи: нема

Повратна вредност: нема

### **GPIO button Saturday**

Опис функције:

Слично функцији за режим рада, мења стање активности радне суботе.

Аргументи: нема

Повратна вредност: нема

### **GPIO button ring**

Опис функције:

Уколико режим рада није активан, служи за ручну репродукцију подразумеване мелодије за активан распоред.

Аргументи: нема

Повратна вредност: нема

### **Initialize**

Опис функције:

Ова функција поставља стање и све променљиве на подразумеване вредности. Такође повива функцију за поставку општенаменских пинова.

Аргументи: нема

Повратна вредност: нема

### **Start**

Опис функције:

Функција која се извржава све време и служи за позивање свих других функција у вези са хардвером. Због тога служи и за постављање активног распореда и покретање функције за звоњење.

Аргументи: нема

Повратна вредност: нема

### **Bell start**

Опис функције:

Позива функцију за звоњење у засебној нити.

Аргументи: распоред звоњења

Повратна вредност: нема

### **Bell active**

Опис функције:

Помоћна функција за упит режима рада.

Аргументи: нема

Повратна вредност: истинитосна вредност режима рада

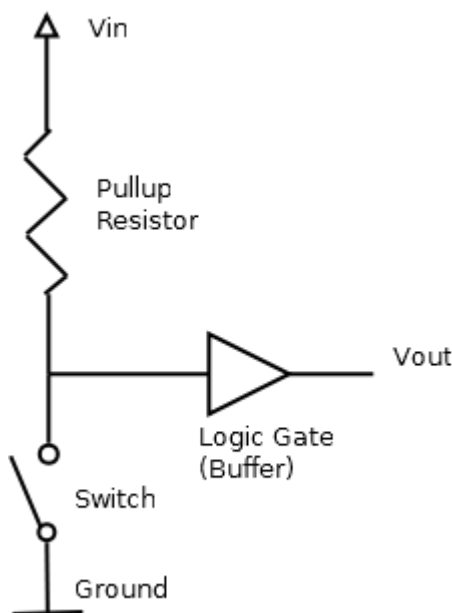
## 4. Хардверска управљачка плоча

Контрола овог система изменом датотеке за подешавања није практична за свакодневну употребу. Промена распореда звоњења, ручно оглашавање и укључивање радне суботе су честе радње за које сувишно користити додатни рачунар. Хардверска управљачка плоча је намењена да дозволи потпуно управљање системом без монитора и тастатуре. На плочи постоји пет светлећих диода и четири тастера. Три светлеће диоде су за приказивање активног распореда, а приказивање режима рада система и радне суботе је уз помоћ две диоде. Први тастер је за промену режима рада, други је за промену распореда, трећи је за укључивање радне суботе, а четврти је за ручно оглашавање.

### 4.1. Тастери

При притискању тастера се не добија савршен прелаз са нула на пет Волти. Због физичких особина проводних металних плочица у тастеру, сигнал узастопно скаче између ова два напона пар милисекунди. При детекцији притиска, потребно је сачекати одређени временски период да се сигнал стабилизује.

Поред тога, уколико је тастер у непритиснутом стању, напон који се добија на излазу није тачно одређен и зависи од спољашњих услова; део тастера није повезан на извор напона и због тога се понаша као антена. Овај проблем се решава додавањем *pull-up* отпорника. Ради се о отпорнику великог отпора који повезује излазни део прекидача са извором напона и кроз који пролази стална слаба струја, дозвољавајући да излазни напон увек буде одређен. На слици испод се налази шематски приказ *pull-up* отпорника.



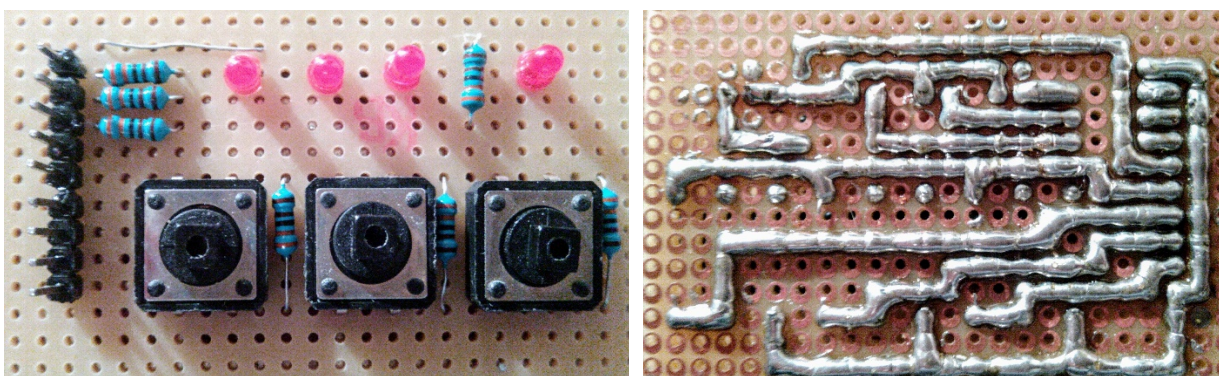
Слика 11 - Шематски приказ *pull-up* отпорника

## 4.2. Развојна плоча

Израда штампаних плоча за електронику је дугачак процес у коме је исправљање грешака скоро немогуће уколико се оне не пронађу у почетку. Због тога је пракса код многих пројеката да се компоненте прво повежу на развојној или прототипној плочи, како би се на време уочиле све грешке.

Постоје две врсте прототипних плоча: за повезивање џампер кабловима и за повезивање лемљењем. У почетку је коришћена плоча за повезивање кабловима због лакших промена у електричном колу, али је касније све пребачено на плочу која се леми.

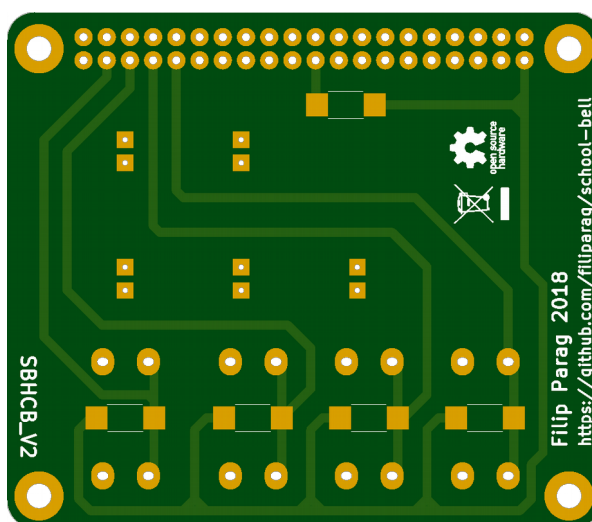
На другу плочу су прво постављени и залемљени хедер пинови, након чега су остале компоненте биле додаване и лемљене. На крају су направљене везе између компоненти, односно жице од лемне жице. На слици испод су приказане обе стране плоче.



Слика 12 - Обострани приказ управљачке плоче

Може се приметити да на плочи нису постављене све компоненте, из разлога да се комплексност лемних жица знатно смањи.

Након успешног тестирања на развојној плочи, у програму КиКед је направљена припрема за штампању плочу. Сама израда штампане плоче је рађена у Кини.



Слика 13 - Илустрација штампане плоче

### 4.3. Повезивање са микрорачунаром

Управљачка плоча је направљена тако да се на микрорачунар може прикључити директно као модул. На овај начин се елиминише потреба за додатним жицама.

У програму постоји провера прикључености управљачке плоче, како би се избегла спонтана активација пинова од стране електромагнетне интерференције.



Слика 14 - Управљачка плоча повезана на микрорачунар

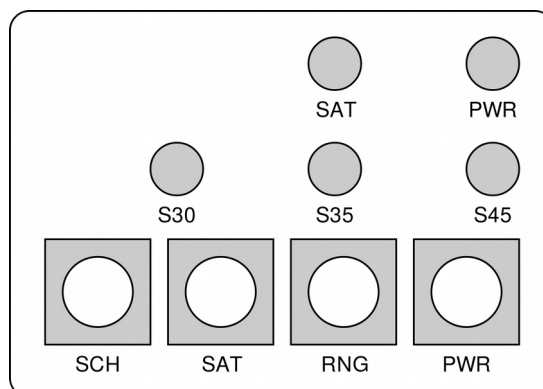
#### 4.4. Упутство за употребу

У првом реду налазе се две светлеће диоде које, гледано с лева, представљају индикатор радне суботе и индикатор укључености.

У другом реду се налазе три светлеће диоде које приказују тренутно активан распоред звоњења. Уколико је активна радна субота, на диодама се приказује распоред за суботу, а у супротном се приказује распоред за све дане.

У трећем реду се налазе четири тастера. Први служи за промену активног распореда звоњења, други за активацију радне суботе, трећи за ручно звоњење, а четврти је за укључивање и искључивање система.

Као и код светлећих диода, тастер за промену активног распореда звоњења мења звоно за радну суботу када је она активна.



##### Укључивање система

Систем се укључује и искључује притиском на тастер *PWR*. Уколико је систем укључен, светлећа диода *PWR* светли, а интеракција са свим осталим тастерима је блокирана све док се систем не искључи. Блокада служи како би се спречиле случајне промене стања система.

##### Режим радне суботе

Режим радне суботе означава да се суботиом извршава посебно одабран распоред звоњења, независно од тога да ли распоред часова за све дане обухвата суботу. На овај начин се током радних дана може унапред подесити систем да у суботу буде скраћена настава. Укључивање и искључивање режима радне суботе се врши тастером *SAT*, а његово је приказано истоименом светлећом диодом.

##### Промена распореда

Тренутно активан распоред се мења притиском тастера *SCH*. На светлећим диодама у другом реду је приказан изабран распоред. Распоред часова са трајањем од 30 минута означава диода *S30*, распоред са часовима од 35 минута диода *S35*, а стандардни распоред са часовима од 45 минута диода *S45*.

Уколико је активан режим радне суботе, све промене тренутно активног распореда утичу на распоред за радну суботу. Уколико је потребно променити распоред за све дане, режим радне суботе се мора привремено искључити.

##### Ручно звоњење

Уколико је потребно ручно покренути звоно ван распореда, то се може урадити притиском на тастер *RNG* док је систем искључен. По притиску се једном репродукује подразумевана мелодија звона за тренутно активан распоред.



## **Закључак**

Од једноставног задатка да се поправи школско звоно у Гимназији „Јован Јовановић Змај“ у Новом Саду, настао је овај напредан систем за озвучење установе који је развијан и ригорозно тестиран две године. У почетку је било много непредвиђених, случајних оглашавања звона, која су углавном била последица покретања система у непознатим условима: од недовољног напајања, нестанака струје и интернета, до тривијалних грешака унутар самог кода. Систем је сада активан већ пар месеци без икаквих сметњи и застоја у раду и поред тога што се у међувремену дограђивала хардверска управљачка плоча.

Током рада на овом пројекту, искусио сам рад на такозваном систему у продукцији, односно у окружењу у којем је свака грешка веома приметна и ремети рад других људи. Били су чести тренуци када сам преко свог рачунара у току наставе морао да анализирам и преправљам код како би се спречила пометња у школској настави.

Развијање овог система је била једна дугачка авантура пуна препрека које нису увек имале очигледан обилазак, али је на крају све свршено по плану; ја сам добио знање и искуство, а школа ново звоно и свакодневно подсећање на једног ђака.

## Садржај ЦД-а

На ЦД-у се налази код за овај систем последњи пут ажуриран у мају 2018. године. Као што је поменуто раније, при покретању ће бити проверена ажурност кода и уколико постоји новија верзија, она ће бити преузета.

Преглед директоријума и датотека на диску:

```
├── config.ini
├── config.py
├── install.sh
├── LICENSE.md
├── macro.py
├── raspberry_old.py
├── raspberry.py
├── ringer.py
├── schedule.py
├── schedules
│   ├── jjzmaj_30.json
│   ├── jjzmaj_35.json
│   └── jjzmaj_45.json
├── school_bell.py
├── sounds
│   ├── bell_long.wav
│   ├── bell_short.wav
│   ├── russian_bells_fadeout.wav
│   ├── russian_bells_long.wav
│   └── russian_bells_short.wav
└── update.sh
```

## Литература

- [1] Чодхери Ш., Шарма Ш., Шарма П., (2014). *Automatic college bell syste*, *International Journal of scientific research and management (IJSRM)*, Морадабад: Институт технологије Масачусетс
- [2] Билић Д., Узуновић Т., (2016) *Embedded automatic scheduling system, 2016 XI International Symposium on Telecommunications (VIHTEL)*, Сарајево: Електротехнички факултет универзитета у Сарајеву
- [3] Охијани Х., Ноње Е., Ибрахим А., (2011) *Design of Microcontroller-Based Automatic School Bell*, Мина: Федерални универзитет технологије Нигерије
- [4] Динеш Кумар Н., Свита Е., Саксена А., (2014) *Implementation Of School/College Bell Using Labview*, Хајдербад: Вигнански институт технологије

## Слике

Слика 1 - Илустрација микрорачунара Разбери Пај 3.....	6
Слика 2 - Графички кориснички интерфејс Разбијана.....	7
Слика 3 - Веб страница за преузимање системских слика.....	9
Слика 4 - Копирање системске слике програмом Ечер.....	10
Слика 5 - Обавештење при првом укључивању Разбијана.....	10
Слика 6 - Графички панел за подешавање система.....	11
Слика 7 - Испис команде за инсталацију програма Гит.....	12
Слика 8 - Инсталација софтвера.....	12
Слика 9 - Упитни прозор за потврду инсталације Тимвјувера.....	13
Слика 10 – Почетни прозор Тимвјувера.....	13
Слика 11 - Шематски приказ <i>pull-up</i> отпорника.....	21
Слика 12 - Обострани приказ управљачке плоче.....	22
Слика 13 - Илустрација штампане плоче.....	22
Слика 14 - Управљачка плоча повезана на микрорачунар.....	23

## БИОГРАФИЈА МАТУРАНТА

Филип Параг је рођен у Суботици 31. августа 1999. године.

Основну школу „Кизур Иштван“ је похађао у Суботици, а 2014. године је у Гимназији „Јован Јовановић Змај“ уписао природно-математички смер. Након завршеног првог полугодишта, преписан је на смер обдарени ученици у рачунарској гимназији.



У осмом разреду основне школе је постао редован полазник Истраживачке станице Петнице, а од 2016. године је члан едукативног рачунарског центра „Тилда Центар“ у Новом Саду.

У Гимназији је основао и води секцију примењене физике и електронике, која за циљ има да ученике упозна са електроником, роботиком и другим областима техничких наука.

Кембриџов сертификат за напредни ниво Енглеског језика, *Cambridge English Advanced* (CAE), је положио у децембру 2015. године.

Као матурант гимназије, планира да настави образовање на Факултету техничких наука Универзитета у Новом Саду.

Датум предаје матурског рада: \_\_\_\_\_

Комисија:

Председник \_\_\_\_\_

Испитивач \_\_\_\_\_

Члан \_\_\_\_\_

Коментар:

Датум одбране: \_\_\_\_\_

Оцена \_\_\_\_\_ ( )