

МАТЕМАТИЧКА ГИМНАЗИЈА

МАТУРСКИ РАД
ИЗ ПРОГРАМИРАЊА

QR кодови

Ученица

Исидора Влајковић, IV_A

Ментор

Коста Ђуришић

Београд, јун, 2025.

Садржај

1	Увод	1
1.1	Шта је QR код	1
1.2	Примене у свакодневном животу	2
1.3	Од Морзеве азбуке до QR кодова: еволуција кодирања информација	3
1.3.1	Морзев код	3
1.3.2	Бар-код	4
1.3.3	Код 49	7
1.3.4	PDF 417	9
1.3.5	QR кодови	10
1.4	Зашто баш QR кодови?	12
2	Како раде QR кодови	13
2.1	Архитектура QR кодова	13
2.1.1	Мирна зона	14
2.1.2	Функционални обрасци	14
2.1.3	Регије информација	15
2.2	Скенирање из више углова и исправљање дисторзије	23
2.3	Корекција грешака	25
2.3.1	Рид-Соломонови кодови	25
3	Додатни аспекти QR кода	33
3.1	Верзије QR кода	33
3.2	Занимљиве примене	36
3.3	Безбедност	42
4	Закључак	43
		43

1 Увод

1.1 Шта је QR код

QR код (енгл. *Quick Response Code* - брз одзив) представља најпознатији тип дводимензионалног бар-кода. Служи за складиштење информација, које се са њега могу прочитати скенером. Најчешће садржи адресу веб-сајта, али може садржати и контакт информације, информације о банковном рачуну, аудио или видео фајл, географску локацију, поруку и слично.

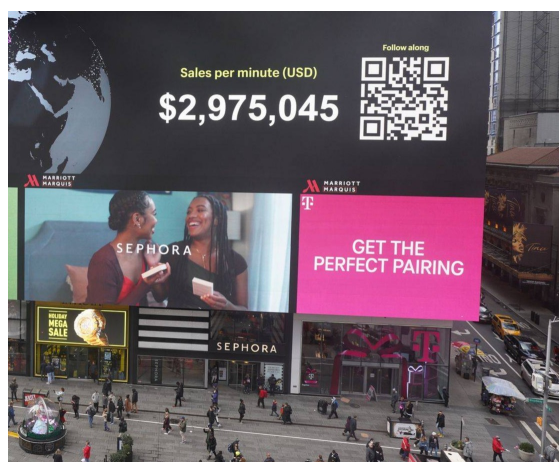
Састоји се из црних и белих квадратића, а препознатљив је по свом позиционом обрасцу тј. трагачу у виду три карактеристична квадрата у угловима (слика 1.1). Они омогућају скенеру лакше детектовање, поравнање и читање QR кода.



Слика 1.1. Пример QR кода с означеним позиционим обрасцем.

1.2 Примене у свакодневном животу

У последњој деценији, QR кодови су значајно добили на популарности јер се лако могу скенирати мобилним телефоном, чиме учествују у бржем преносу и размени информација у свакодневном животу. Практичност њихове примене је довела до тога да се данас могу наћи готово свуда - на фискалним рачунима, амбалажи производа, бизнис картама, картама за концерте или јавни превоз, рекламама и билбордима (слика 1.2а), јеловницима у ресторанима и кафићима (слика 1.2б), током плаћања телефоном (слика 1.2в), у хотелима за приступ интернету (слика 1.2г) итд.



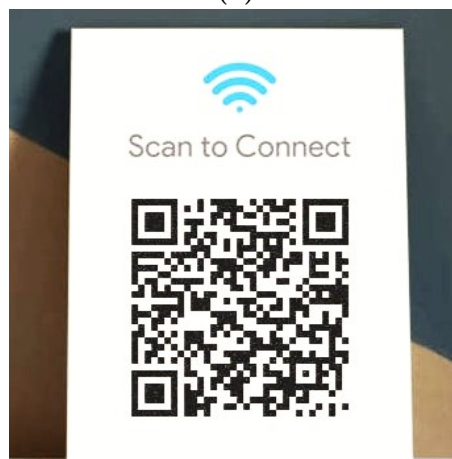
(а)



(б)



(в)



(г)

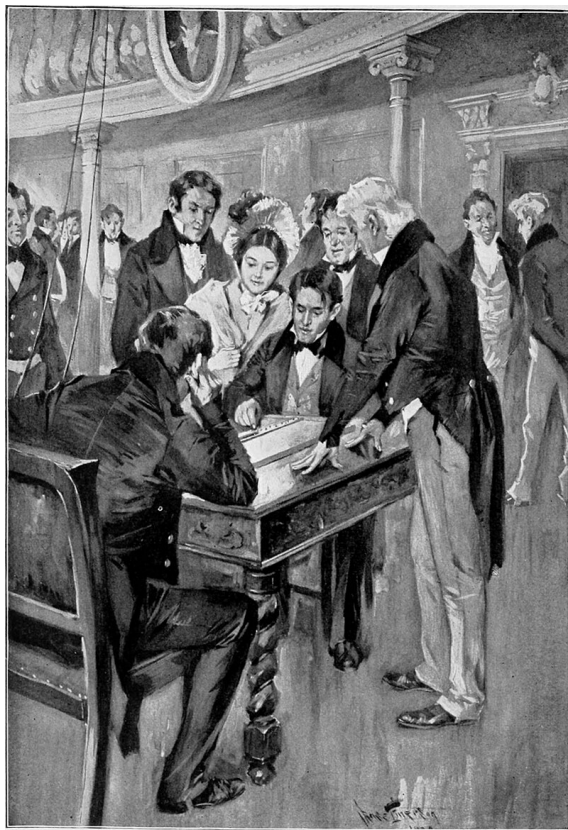
Слика 1.2. Примери QR кодова из свакодневног живота.

1.3 Од Морзеве азбуке до QR кодова: еволуција кодирања информација

1.3.1 Морзев код

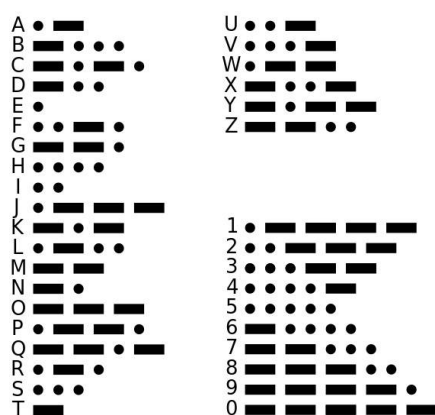
Године 1825, познати амерички сликар живео је са супругом и двоје деце у Њу Хејвену у америчкој држави Конектикат, када је позван да наслика портрет маркиза де Лафајета, хероја америчке револуције. Иако је његова супруга исчекивала рођење њиховог трећег детета сваког тренутка, прилика је била исувише добра да би је могао одбити, па је убрзо кренуо за Вашингтон. Једног дана, добио је писмо у коме је писало да му је жена болесна након порођаја. Забринут, путовао је коњем и колима, дању и ноћу, али било је прекасно. Његова жена је већ била преминула, а још је и пропустио сахрану. Сликара се звао Семјуел Ф. Б. Морзе (енгл. *Samuel Finley Breeze Morse*). Од тада је Морзе одлучио да пронађе бржи начин комуникације на велике удаљености.

Замисао о електричном телеграфу му је пала на памет 1832. године, након што је начуо разговор о новооткривеном електромагнету, на броду у повратку са студија уметности у Европи. Похађао је предавања о електрицитету, области која се у то време брзо развијала, на Универзитету у Њујорку где је добио посао као професор уметности. Уз помоћ моћних електромагнета физичара Џозефа Хенрија, као и материјала и радне снаге набављене од стране механичког инжењера Алфреда Вејла, осмислио је машину која је могла да шаље електричне импулсе дуж жице. Прва телеграфска линија у Сједињеним Америчким Државама, од Балтимора до Вашингтона, завршена је 1844. године, а 24. маја је послат и први телеграм (слика 1.3), с поруком „Шта је учинио Бог” (енгл. *What hath God wrought?*), преузетом из Библије (Четврта књига Мојсијева, 23:23).



Слика 1.3. Семјуел Морзе шаље први телеграм, илустрација из 1900. год.

Ово није био први електрични телеграф, али је био најједноставнији. У Великој Британији, други тим је поставио низ кола која су померала пет магнетних игала које би показивале на слова и бројеве. Морзев систем је захтевао само једно коло, али једноставност апарата је захтевала паметнији метод шифровања информација. На колу је било могуће слати кратке или дуге импулсе. Морзе их је претворио у тачке и цртице. Најчешће коришћена слова могла су се послати једним притиском на тастер, тачка за Е и цртица за Т. Остала слова су била распоређена по фреквенцији и додељени су им све сложенији кодови. Тако је настала Морзеова азбука (слика 1.4).



Слика 1.4. Интернационални морзев код.

дуго коришћен у војсци, поморским комуникацијама и авијацији. Морзеова азбука је револуционисала комуникацију, али у следећем веку ће трансформисати потпуно другачију индустрију.

Морзев оригинални телеграфски пријемник користио је механички сатни механизам за померање папирне траке, на којој је оловка, покренута механизмом активираним од стране електромагнета, правила удубљења и исписивала текст морзевом азбуком. Убрзо су оператери схватили да могу да препознају слова само по звуку. За дуги знак је узето да траје као три кратка. Пауза између знакова траје као један кратак, између речи као један дуг, док је између реченица седам кратких. Ово је убрзало брзину слања и примања информација, па је Морзеова азбука постала међународни стандард за брзо слање порука,

1.3.2 Бар-код

Крајем четрдесетих година двадесетог века, Бернард Силвер је био студент инжењерства на Универзитету Дрексел у Пенсилванији. Једног дана је чуо председника локалног ланца супермаркета како тражи од декана факултета да пронађе начин да убрза процес плаћања. У то време, благајници су морали ручно да уносе цену сваког производа, процес толико заморан да су многи развили синдром карпалног тунела. Силвер је испричао свом пријатељу Норману Џозефу Вудланду о том проблему и заједно су 1948. године почели да развијају идеју о каси која би могла да чита ознаку производа и аутоматски унесе тачну његову цену. Њихова прва идеја је била да ознаке буду обрасци од мастила које светли под ултраљубичастим светлом, али су наишли на проблеме попут нестабилности мастила и вели-

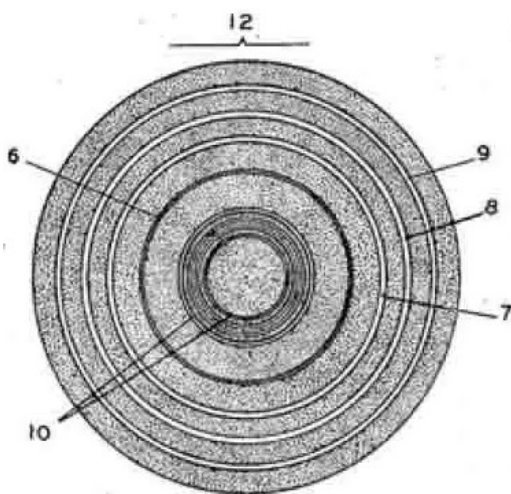
ких трошкова штампања. Након неколико неуспелих покушаја, Вудланд се преселио на Флориду. Једног дана на плажи је нацртао у песку неколико тачака и цртица Морзеве азбуке, нешто што му је било веома познато јер је био извиђач као дете. „Само сам продужио тачке и цртице надоле и од њих направио уске и широке линије”, рекао је Норман (слика 1.5).

Тако је настао први бар-код. Проблем је настао када су Вудланд и Силвер схватили да је скенирање новог кода успешно само ако је он усправан. Њихово решење је било да уске и дебеле линије претворе у концентричне кружнице, тако да код личи на мету, што би омогућило његово читавање из било ког угла (слика 1.6).



Слика 1.5. Сигнал за помоћ СОС приказан у виду Морзеве азбуке и бар-кода.

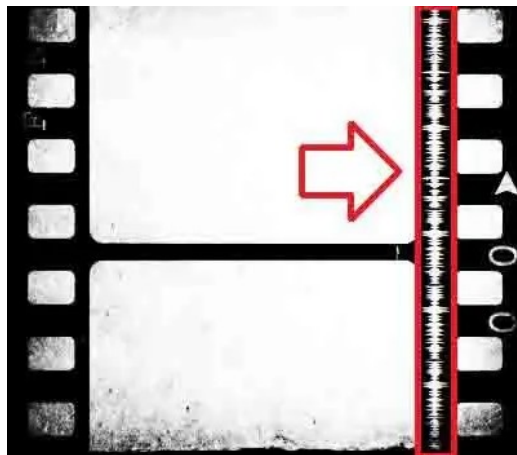
Следећи изазов је био направити скенер за бар-код. Први прототип је био величине стола и састојао се од два кључна елемента: сијалице са жарном нити од 500 вати као извора светлости и фотомултипликаторске цеви, дизајниране за системе филмског звука, као читача. Инспирацију за коришћење фотомултипликаторске цеви су добили из филмске индустрије. У то време, звук се додавао филмовима тако што се слика звучног таласа стављала директно на ивицу филмске траке (слика 1.7). На супротној ивици траке, налазио се фотомултипликатор. Он је честице светлости (фотоне), коју је емитовао пројектор на филмско платно, претварао у електроне. Како се филмска трака померала, мењали су се и фотони пројектовани кроз слике звучних таласа, а самим тим и ток електрона, што би стварало електрични сигнал, који би даље био претворен у звук путем појачала и звучника.



Слика 1.6. Кружни бар-код. Линије 6, 7, 8 и 9 су мање рефлективне него оне означене бројем 10.

Вудланд и Силвер су прилагодили ову технологију тако што су усмерили извор светлости ка бар-коду, чије су беле линије одбијале више светлости него црне. Одбијену светлост детектује фотомултипликатор. Уместо звука, електрични сигнал би био претворен у текст. Овај изум био је још увек далеко од практичне примене. У то време, компјутери су били величине три фрижидера, а она сијалица од 500 вати је била заслепљујуће јака, али скупа и енергетски

неефикасна. Био им је потребан извор који фокусира велику количину светлости на мали простор, али ласери још увек нису постојали 1952. године.



Слика 1.7. Слика звучног таласа на ивици филмске траке.

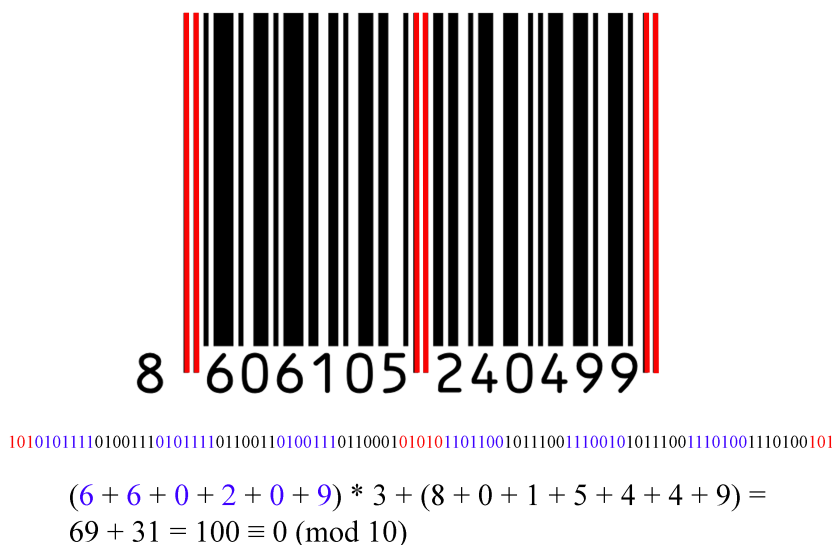
Коначно, 1972. године, супермаркет у Синсинатију је одлучио да испроба кружни бар-код. Нажалост, тест је био неуспешан због проблема са штампањем кода - мастило би се размазало у смеру кретања папира. Вудланд је приметио да линеарни бар-код нема тај проблем јер се штампа у истом смеру као и пруге. Сваки вишак мастила стиче на врху или дну и никакве информације се не губе. Ипак, и даље је остао проблем да се линеарни бар-кодови могу скенирати само из једног смера. Ову последњу препреку Вудланд је решио уз помоћ колеге из ИБМ-а (енгл. *International Business Machines Corporation*), креирањем УПЦ бар-кода (енгл. *Universal Product Code* типа А.

Он има 12 цифара и данас се користи већином у САД-у и Канади, док је у Европи много чешћи њему врло сличан, и с њим компатибилан, следбеник, бар-код ЕАН-13 (енгл. *European Article Number*), који има једну цифру више из потребе да се прецизира из које државе потиче. Стандард форматирања ЕАН-13 (слика 1.8) је следећи:

1. Свака цифра, осим прве, састоји се из два бела размака, који се кодирају нулама, и две црне траке, које се кодирају јединицама. Такође, свака цифра је дужине 7 битова.
2. По један пар црних вертикалних линија, кодиран као „101”, је постављен на почетку и крају бар-кода, како би скенер могао правилно да га прочита. На средини се налази још један пар, кодиран као „01010”, који дели бар-код на леву и десну страну. На левој страни, цифре почињу нулом, а на десној јединицом. На овај начин, скенер зна у ком смеру да чита бар-код, чак и ако је окренут наопачке.
3. Прве 3 цифре представљају шифру државе порекла произвођача. Понекад се једна цифра изостави ако је нула. Шифра за Србију је 860.
4. Остале 4 (односно 5) цифре на левој страни служе за код произвођача.
5. Затим следи 5 цифара за код производа.
6. Последња цифра служи за осигуравање да је бар-код правилно прочитан. Ако саберемо цифре на парним позицијама, помножимо збир са 3, додамо том броју збир цифара на непарним позицијама и све то поделимо са 10, добићемо неки остатак. Ако је остатак нула, последња цифра је нула, а у супротном је

број 10 умањен за остатак. Уколико скенер не може да прочита једну цифру бар-кода, може да је срачуна уз помоћ последње цифре. Један од разлога за то што цифре на парним позицијама имају већу „тежину” од оних на непарним јесте да се често дешава замена суседних цифара приликом скенирања или личног уношења цифара.

7. Друга цифра, као и цела десна страна бар-кода, увек су непарне парности, што значи да је збир јединица у сваком од њихових кодова непаран. Парност осталих цифара зависи од прве цифре. На овај начин се одређује вредност прве цифре.



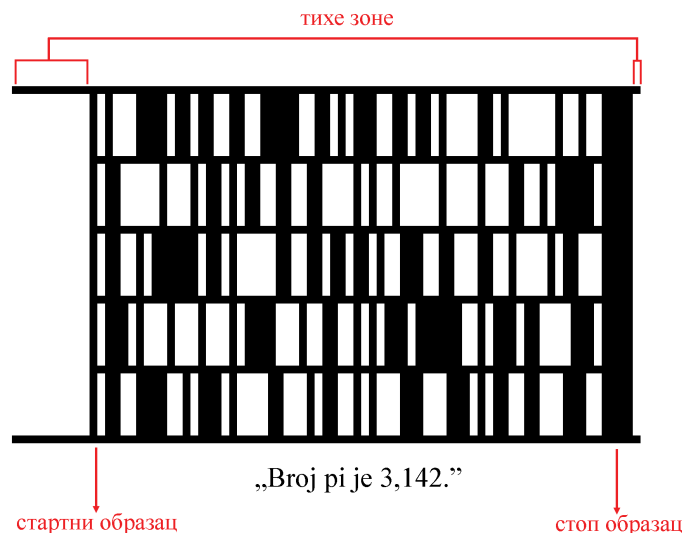
Слика 1.8. Бар-код преузет с паковања Fresh марамица с камилицом. Црвеном бојом су обележени парови линија раздвајања, а плавом парне цифре бар-кода.

Постоји трилион различитих комбинација, а до данас је регистровано 1,3 милијарде бар-кодова. Тај број расте сваки дан, тако да нам неће трајати заувек. Ипак, то није разлог зашто су тражене алтернативе бар-кодовима, него количина информација које он може да садржи.

1.3.3 Код 49

У Великој Британији, стока је 1986. године почела да одуми́ре од болести лудих крава, неизлечиве болести која уништава мозак. Људи би били заражени уколико поједу говедину која садржи ткиво мозга или кичмене мождине заражене краве. Пошто ниједан тест није могао да открије болест код живих крава, милиони су убијени, па је тако започета потрага за методом праћења извора увоза говедине. Са свим информацијама које би ово захтевало за било који комад говедине, бар-кодови нису били довољни. Амерички

проналазач Дејвид Алејс (енгл. *David Allais*) покушао је да реши проблем слагањем много бар-кодова један на други. Резултат, Код 49, изгледао је као полица за књиге (слика 1.9). Његове значајне особине биле су велика густина података, као и способност да садржи било који од 128 карактера из ASCII табеле (енгл. *American Standard Code for Information Interchange*).



Слика 1.9. Обележени пример кода 49.

расцем „1111”, како би скенер разликовао леву и десну страну. Између њих, налази се 8 карактера, где последњи служи за проверу исправности датог реда, на основу алгоритма који користи модул броја 49.

Постоји и карактер који проверава исправност целог кода и његов алгоритам користи модул броја 2401 (односно 49^2). Карактери могу бити различитих парности, у зависности од реда у ком се налазе. Тако се одређује број реда приликом читања. Сваки од ASCII карактера се добија или директно из табеле на слици 1.10, или тако што се испред неког члана табеле стави нека од следећих кодних речи (енгл. *codeword*): Shift1, у ознаци S1 и Shift2, у ознаци S2. На пример, мало слово а изгледа као S2 А, док је узвичник S1 6.

Са леве и десне стране кода 49, налазе се такозване „тихе зоне”, празан простор који омогућава скенеру да разликује бар-код од околине и самим тим га лакше детектује. Сам код 49 има између 2 и 8 редова, који се састоје из 18 црних трака и 17 белих простора. Редови су раздвојени хоризонталним линијама. Сваки ред почиње стартним обрасцем кодираним као „100”, а завршава стоп об-

вредност	карактер	вредност	карактер
0	0	25	P
1	1	26	Q
2	2	27	R
3	3	28	S
4	4	29	T
5	5	30	U
6	6	31	V
7	7	32	W
8	8	33	X
9	9	34	Y
10	A	35	Z
11	B	36	Hyphen (-)
12	C	37	Period(.)
13	D	38	Space Symbol
14	E	39	Dollar sign (\$)
15	F	40	Slash mark (/)
16	G	41	Plus sign (+)
17	H	42	Percent sign (%)
18	I	43	S1 (Shift 1)
19	J	44	S2 (Shift 2)
20	K	45	FNC1 (Function 1)
21	L	46	FNC2 (Function 2)
22	M	47	FNC3 (Function 3)
23	N	48	(Numeric Shift)
24	O		

Слика 1.10. Табела са кодним речима и одговарајућим вредностима. Кодних речи у табели има 49, по чему је код 49 добио име.

1.3.4 PDF 417

Из кода 49 је настао, данас много чешће коришћен, PDF 417, који се налази на авионским картама (слика 1.11). Настао је 1991. године. Реч „PDF” у називу је скраћеница од *Portable Data File* (Преносива датотека података) и нема везе са PDF фајлом, који је настао касније; „417” означава да се сваки образац у коду састоји од 4 црне траке и 4 бела размака у обрасцу, укупне дужине 17.

PDF 417 има по тиху (мирну) зону с леве и десне стране, почиње стартним (почетним) обрасцем, а завршава се стоп обрасцем (обрасцем заустављања), као што је приказано на слици 1.12. Може бити висине 3 до 90 редова. Сваки ред има два индикатора. Леви стоји на почетку и садржи информације о броју и нивоу корекције грешака тог реда, док десни стоји на крају и садржи додатне информације. За корекцију грешака користи се Рид-Соломонов алгоритам, о ком ће бити речи у наредној глави. Ниво корекције грешака (енгл. *error correction level*) означава количину додатних кодних речи које се користе за откривање и исправљање грешака у подацима, омогућавајући тачно читавање чак и када је PDF417 делимично оштећен. Што више додатних кодних речи има, то је већи део података могуће повратити. Доступна су четири нивоа корекције грешака која се могу изабрати, што омогућава читање бар кода чак и ако је једна половина кода оштећена. PDF417 који има корекцију грешака подешену на минимални ниво, подржава до 1850 ASCII карактера (8-битни систем), 2710 цифара или 1108 бајтова. PDF417 симболи се чак могу повезати са другим симболима који се скенирају у низу, што омогућава чување још више података.



Слика 1.11. PDF 417 на авионској карти.



Слика 1.12. Обележени пример PDF417 симбола

Дакле, значајно је повећана количина информација који један симбол може

да садржи. Ипак, површина PDF417 симбола је била превелика, а брзина скенирања преспора за већину индустрија. Ускоро је показано да се, за исту количину садржаја, простор може смањити и до четири пута, а брзина скенирања и до пет.

1.3.5 QR кодови

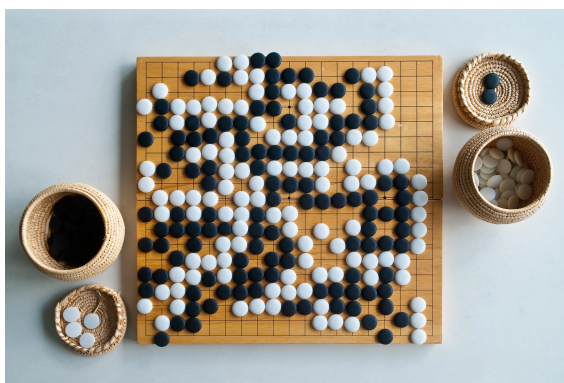


Слика 1.13. Масахиро Хара, отац QR кода.

Масахиро Хара (слика 1.13), инжењер у јапанском произвођачу ауто-делова, Денсо Вејв (енгл. *DENSO Wave*), подружници корпорације Тојота, био је фрустриран због потребе да скенира више бар-кодова за попуњавање једне кутије са деловима аутомобила. „На свакој кутији је било више од десет бар-кодова, а како се уље користи у фабрикама, бар-кодови би се често задрљали и постали нечитљиви, па смо морали да уносимо податке на тастатури”, рекао је Хара. „Запослени су се уморили од скенирања истих кутија више пута, и то нас је довело до тога да смислимо код који би омогућио пренос велике количине информација једним скенирањем.”

Идеја за првобитни дизајн наизменичних квадрата који је представио тим истраживача, на челу са Масахиром Харом, потекао је из игре Го, кинеске мисаоне игре за два играча. Го се игра тако што се црно и бело округло камење поставља на квадратну таблу с хоризонталним и вертикалним линија (слика 1.14).

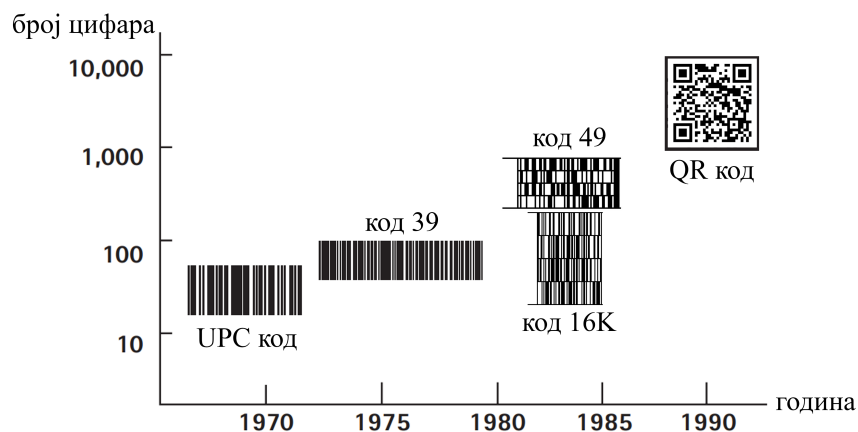
Циљ игре је заузети већу територију од противника. Игра је стара између 3 и 4 хиљаде година, што је чини најстаријом игром која се до данас игра по неизмењеним правилима. „Раније сам играо Го током паузе за ручак. У Гоу, постављате камење на пресеке линија, али чак и ако је камен мало померен, ми људи и даље видимо где је постављен. Дакле, док сам играо Го, размишљао сам како би се ова структура могла прочитати и одатле ми је пала на памет идеја за црно-белу матрицу”, рекао је Масахиро Хара. Током 1994. године, Денсо Вејв објављује QR код. Као резултат Масахириног



Слика 1.14. друштвена игра Го.

труда и маркетинга корпорације, QR кодови су почели да се користе у аутомобилској индустрији за системе контроле производње. Првобитна сврха QR кода била је да олакша праћење врста и бројева аутомобилских делова, заменом појединачно скенираних бар-код етикета на свакој кутији ауто-делова једном етикетом која је садржала податке сваке етикете. Ипак, примена QR кодова се јако брзо раширила, почевши од коришћења од стране великих компанија прехранбене индустрије, фармацеутских компанија и компанија за производњу сочива, са циљем праћења робе. Захваљујући подршци водећих компанија из различитих индустрија, QR кодови су ускоро постали распрострањени широм Јапана, а већ 2002. године телефони са могућношћу читања QR кодова су се појавили. Данас су скенери QR кодова директно уграђени у апликацију камере мобилних телефона (још од 2017. године). У 2022. години, 89 милиона људи у САД-у је скенирало QR код користећи своје мобилне уређаје, 26% више у односу на 2020. годину. Тај број расте сваког дана.

Употреба QR кодова далеко је премашила очекивања Харе и његових колега. Хару је највише изненадило то што се QR кодови сада користе за електронска плаћања. „Никада нисам мислио да ће се (QR кодови) користи-



Слика 1.15. Еволуција симбола за кодирање информација.

ти за размену новца када смо развијали систем. И даље сам забринут да би нешто могло поћи страшно по злу”, рекао је кроз смех.

Масахиро Хара још увек ради на побољшању QR кода, као генерални директор одељења за инжењеринг у Денсо Вејву. Рекао је да након пензионисања жели да се бави пољопривредом и развија нове и укусније сорте воћа.

Данашњи QR код може да садржи и до 3 килобајта података, односно преко 7 хиљада цифара, што је неколико стотина пута више него први бар-кодови. Да су постојали 1969. године, било би потребно само 26 QR кодова верзије 40 за чување свих информација потребних компјутеру Апола 11 за слање људи на Месец. На слици 1.15 приказан је преглед развоја машински читљивих кодова кроз време, са фокусом на количину информација коју сваки симбол може садржати. На графику се може уочити јасан раст капацитета симбола кроз године, с QR кодом као великом прекретницом.

1.4 Зашто баш QR кодови?

Исте године када се појавио QR код, изумљени су и други, слични матрични кодови, као што су Data Matrix и VeriCode. Зашто се баш QR код издвојио и постао популаран међу ширим масама?

Data Matrix симбол (слика 1.16) је врло ефикасан за означавање малих индустријских компоненти због изузетно малих димензија и користи се претежно у логистици, електроници и здравству. Ипак, његова видљивост ван тих сектора је ограничена - камере мобилних телефона тешко могу распознати ситне делове Data Matrix-а. Такође, читачу је много теже да детектује позициони образац Data Matrix-а у облику латиничног слова Л, за разлику од врло упечатљивог који QR код има, што успорава скенирање.



Слика 1.16. Data Matrix симбол.

VeriCode симбол (слика 1.17) је био коришћен од стране Насе за праћење и идентификацију делова спејс шатлова. Он представља власнички формат са затвореним стандардом, који се користи углавном у високо контролисаним индустријама попут фармације и производње, што ограничава његову ширу употребу. С друге стране, QR код је отвореног формата, није заштићен ауторским правима и бесплатан је за коришћење, што је подстакло његово брзо усвајање. Када су 2012. године QR кодови освојили награду у категорији Медија за Индустрију за „Добар дизајн“, Масахиро Хара је у говору на додели награде објаснио да је његова жеља да се његова креација развија неvezано од њега и да је зато и одлучио да не примењује патентна права.



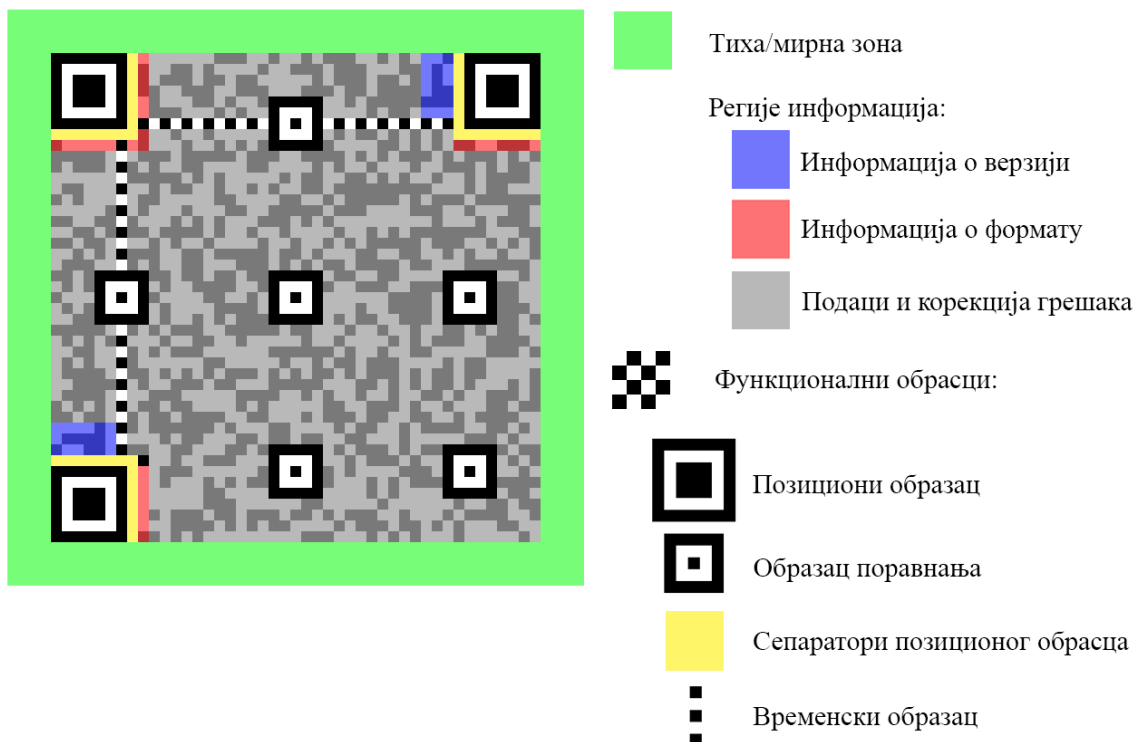
Слика 1.17. VeriCode® simbol.

2 Како раде QR кодови

2.1 Архитектура QR кодова

QR код је симбол матричног типа са компонентама организованим у квадратну архитектуру. Састоји се из црних (тј. тамних) и белих (тј. светлих) квадратића тј. модула, распоређених унутар квадратне мреже. Модул је најмањи квадратни елемент од ког је састављен QR код. Сваки модул представља један бинарни податак тј. бит.

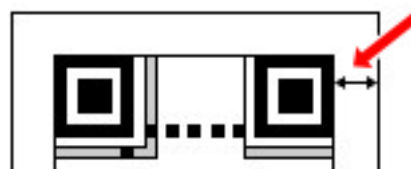
QR код је сачињен из више области, свака са јединственом функцијом унутар архитектуре (слика 2.1).



Слика 2.1. Пример QR кода верзије 7 с обележеним деловима.

2.1.1 Мирна зона

Мирна, односно тиха зона (енгл. *Quiet Zone*) је празна маргина дужине 4 модула, која окружује QR код са свих страна. Она је неопходна за читање QR кода јер помаже читачу да јасно разграничи код од површине на којој се он налази.



Слика 2.2. Мирна зона.

2.1.2 Функционални обрасци

Позициони образац

Позициони образац (енгл. *Position Detection Pattern* или *Finder Pattern*), такође познат и као трагач, састоји се од три велика квадрата у горњем-левом, горњем-десном и доњем-левом углу QR кода. Идентификовањем позиционог обрасца, читач може брзо да одреди позицију QR кода, што је неопходно за прецизно декодирање и скенирање. Доследно постављање ових шаблона осигурава да се QR кодови могу скенирати из више углова и под различитим условима осветљења.

На спољашњим ивицама позиционог обрасца се налази **сепаратор**, бели простор чија је функција одвајања позиционог обрасца од самог садржаја кода.

Образац поравнања

Образац поравнања (енгл. *Alignment Pattern*) чине мањи квадрати, слични онима из позиционог обрасца. Он омогућава читачу да коригује изобличења слике (нпр. ако је код постављен на закривљену површину). Релативна величина и растојање квадрата обрасца поравнања у односу на квадрате позиционог обрасца омогућава софтверу да QR код преобликује у одговарајући квадрат.

Број и распоред квадрата обрасца поравнања расте са сложеностју кода. Прва верзија QR кода нема ниједан, наредних неколико верзија има само један, верзија бр. 10 има шест, док четрдесета верзија (последња верзија) има чак 35.



Слика 2.3. А: образац поравнања.

Временски образац

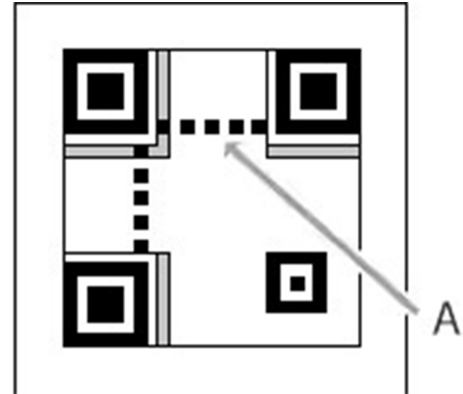
Временски образац (енгл. *Timing Pattern*) чине две нормалне линије с наизменично постављеним црним и белим квадратићима. Хоризонтална линија

овог обрасца је постављена на шестом реду QR кода између сепаратора, а вертикална линија је смештена на шестој колони између сепаратора.

Њихова сврха је да дефинишу координатни систем унутар мреже и помогну читачу у одређивању величине QR кода, као и у прецизном лоцирању модула (тј. налажењу њихових координата). Временски образац делује као референтна мрежа, осигуравајући да се модули читају исправно и доследно.

На основу временског обрасца се може закључити која верзија QR кода је у питању.

Ако се линија састоји из по 5 црних и белих квадратића, у питању је прва верзија; друга их има 9, а четрдесета 81.

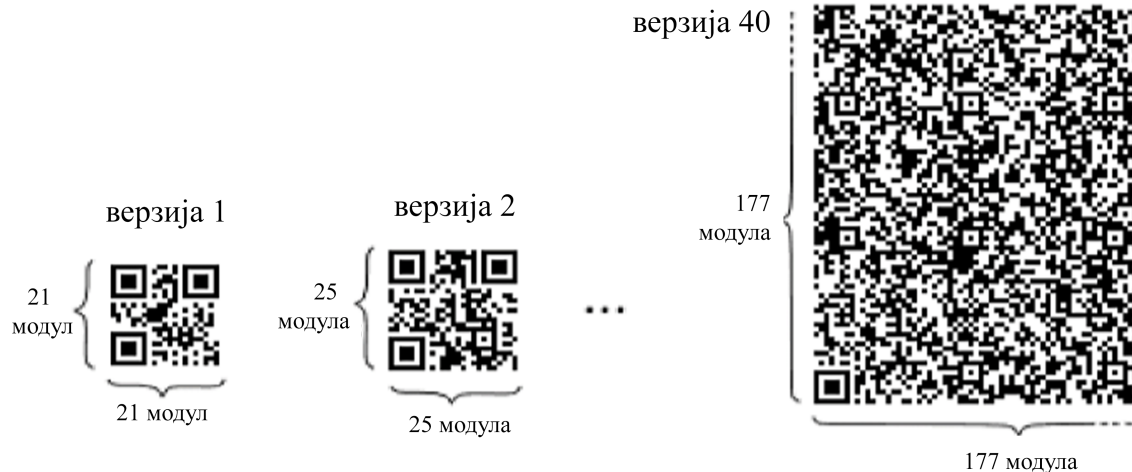


Слика 2.4. А: временски образац.

2.1.3 Регије информација

Информације о верзији

Код QR кодова верзије 7 и више, додатни модули указују на верзију кода, што је важно због сложености њихове унутрашње структуре. Постоји 40 верзија QR кода. Димензије прве верзије су 21×21 модула (слика 2.5. Свака следећа верзија је дужа и шири за по 4 модула. Дакле, верзија 40 биће димензија 177×177 , као на слици 2.5 ($21 + 39 \cdot 4 = 177$).



Слика 2.5. Верзије QR кода.

Област информација о верзији чине два правоугаоника димензија 6×3 модула, један у доњем-левом (6 модула широк, а 3 висок), а други у горњем-

десном углу (3 модула широк, а 6 висок). Они стоје поред позиционог обрасца, без обзира на величину QR кода. Стринг информација о верзији је дужине 18 бита, од којих шест кодира број верзије QR кода, а 12 служи за корекцију грешака. За одређивање стринга користи се следећи полином: $x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^5 + x^2 + 1$. Можемо га представити као бинарни стринг 1111100100101.

Прво направимо шестобитни бинарни низ који представља број верзије. Узмемо за пример код верзије 7. Шестобитни бинарни еквивалент броја 7 је 000111. Претворимо ово у 18-битни низ тако што са десне стране допунимо нулама:

000111000000000000

Затим уклонимо нуле са леве стране:

1110000000000000

Сада, допунимо полином са десне стране нулама да би био исте дужине:

111110010010100

Након тога извршимо XOR операцију над стрингом верзије и полинома: $1110000000000 \hat{=} 111110010010100 = 110010010100$.

Овај низ је већ потребне дужине од 12 - да је већи, било би потребно даље дељење. Уколико је резултат краћи од 12 битова, само бива попуњен нулама с леве стране. Овине смо добили низ за корекцију грешака. На њега додајемо с леве стране шестобитни низ верзије, чиме добијамо крајњи стринг информације о верзији: 000111110010010100. Оба правоугаоника из области информација о верзији попуњавамо с истим информацијама, како би један био резервни у случају да други бива оштећен. Попуњавамо их битовима из стринга на следећи начин:

0	3	6	9	12	15
1	4	7	10	13	16
2	5	8	11	14	17

У горњем-левом делу правоугаоника се налази бит најмање тежине (скроз десни бит стринга), а у доњем-десном бит највеће тежине (скроз леви бит стринга).

Информације о формату

Ови подаци се налазе поред сепаратора и садрже информације о нивоу корекције грешака и примењеној шеми маскирања, што је неопходно за правилно декодирање садржаја QR кода.

У информације о формату не спада један модул, заокружен црвеном бојом на

слици 2.6. Он је увек црн, а разлог за његову боју не постоји. Масахиро Хара

је рекао да је просто морао да бира између беле и црне. Сматра се да је сврха овог модула да означи доње-леви позициони квадрат. Његове координате су $(8, 4 \cdot \text{број верзије} + 9)$. QR кодови користе четири нивоа корекције грешака. Нивои су означени као L, M, Q и H, при чему сваки представља различит степен отпорности: L (енгл. *Low*) – омогућава исправку до 7%;

M (енгл. *Medium*) – до 15%;

Q (енгл. *Quartile*) – до 25%;

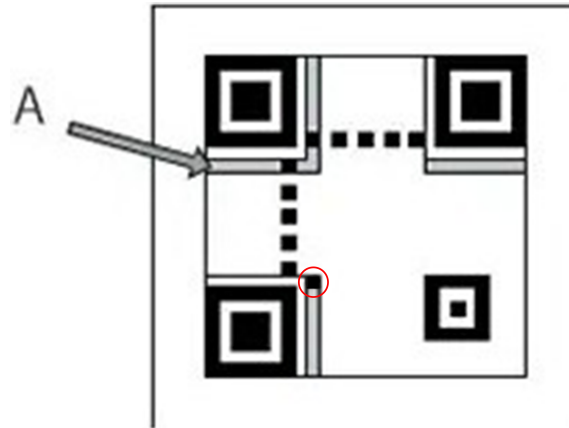
H (енгл. *High*) – до 30% оштећених података.

Избор нивоа зависи од конкретног случаја употребе – виши ниво пружа већу отпорност на оштећења, али смањује капацитет за корисничке податке, јер више простора заузима заштитни код. Коришћење нижег нивоа корекције грешака омогућава постављање слике или логоа на средину QR кода, што компаније често раде како би учиниле QR код препознатљивим и примамљивијим. У пракси се често бира компромис између поузданости и количине информација. Стринг информације о формату је дугачак 15 битоа. Састоји се из стринга од пет битоа који кодира ниво корекције грешака (2 бита) и шаблон маске (3 бита) који се користи у QR коду. Осталих 10 битоа служе за корекцију грешака, и добијају се на сличан начин као за информације о верзији, коришћењем претходно добијених пет битоа и полинома $x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1$.

Код за нивое корекције грешака је следећи:

Ниво корекције грешака	Битови	Вредност
L	01	1
M	00	0
Q	10	2
H	11	3

Процес маскирања представља прераспodelу модула главног садржаја QR кода (података и кода за корекцију грешака), тако да буду равномерније



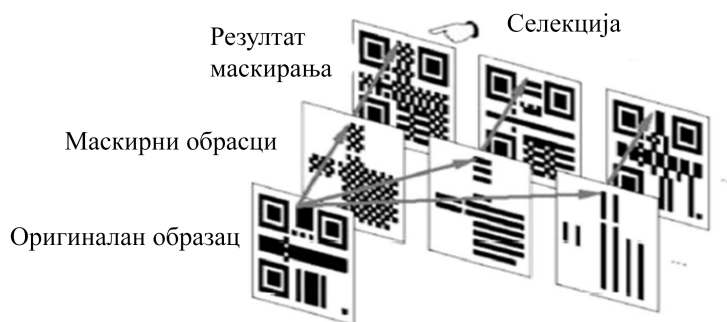
Слика 2.6. А: информације о формату.

распоређени тј. да се чини да су црни и бели квадратићи поређани више „насумично”. Другим речима, циљ је да изолованих, великих, једнобојних површина буде што мање, како би превођење прочитаног садржаја у бинарни систем било што ефикасније. Процес се врши тако што се изврши XOR операција над садржајем и обрасцем маске (Слика 2.7). Постоји 8 врста маскирних образаца (x и y су координате модула):

Битови	Маскирни образац	
111	$x \bmod 3 = 0$	
110	$(x + y) \bmod 3 = 0$	
101	$(x + y) \bmod 2 = 0$	
100	$y \bmod 2 = 0$	
011	$((xy \bmod 3) + xy) \bmod 2 = 0$	
010	$((xy \bmod 3) + x + y) \bmod 2 = 0$	
001	$(x/2 + y/3) \bmod 2 = 0$	
000	$(xy \bmod 2) + (xy \bmod 3) = 0$	

На пример, прва маска у табели промениће боју сваком трећем модулу у реду QR кода.

На садржај QR кода ће бити примењена свака од осам маски. Затим ће сваки добијени образац бити процењен на основу тога колико је „шарен”, а образац са најмање казнених поена, заједно са резултатом XOR калкулације ће бити ускладиштен у област са информацијама о садржају. Казнени поени се додељују за сваку групу од пет или више модула исте боје у реду/колони, област 2×2 модула исте боје у матрици, образац који подсећа на позициони, као и за процентуалну разлику између црних и белих модула (треба да буде што мања).



Слика 2.7. Процес маскирања QR кода.

Садржај: подаци и корекција грешака

QR код има четири режима кодирања. Сваки режим има четворобитни индикатор који га идентификује. Кодирани подаци морају почети одговарајућим индикатором режима јер он одређује тип података који следе. Следећа табела наводи индикаторе режима и максималан број карактера тог типа који QR код може садржати:

Назив режима	Индикатор режима	Максимални број карактера који 40-L код може да садржи у овом режиму
Нумерички режим	0001	7089
Алфанумерички режим	0010	4296
Бајтови	0100	2953
Канџи (јапанско писмо)	1000	1817

Индикатор броја карактера је низ битова који представља број карактера који се кодирају. Он мора бити постављен након индикатора режима. Штавише, мора бити дугачак одређени број битова, у зависности од верзије QR кода. У нумеричком режиму, на пример, за верзије 1-9 мора бити дугачак 10 битова, за верзије 10-26 је 12 битова, а за остале верзије 14.

Затим следи кодирање поруке. Ако је низ битова поруке краћи од укупног броја потребних битова, на десну страну низа мора се додати терминатор од највише четири нуле. Пример за верзију 1 QR кода - нека је наша порука „Hello World”:

Индикатор режима: 0010

Индикатор броја карактера: 000001011

Кодирана порука: 01100001011 01111000110 10001011100 10110111000 10011010100 001101

Терминатор: 0000

Након додавања терминатора, ако укупан број битова у низу није дељив са 8, потребно је додати одговарајући број нула с десне стране. У нашем примеру, потребно је додати две нуле: 00100000 01011011 00001011 01111000 11010001 01110010 11011100 01001101 01000011 01000000

Уколико стринг и даље није довољно дугачак да попуни максимални капацитет QR кода, додају се следећи бајтови на крај стринга, понављајући поступак док стринг не достигне максималну дужину: 11101100 00010001

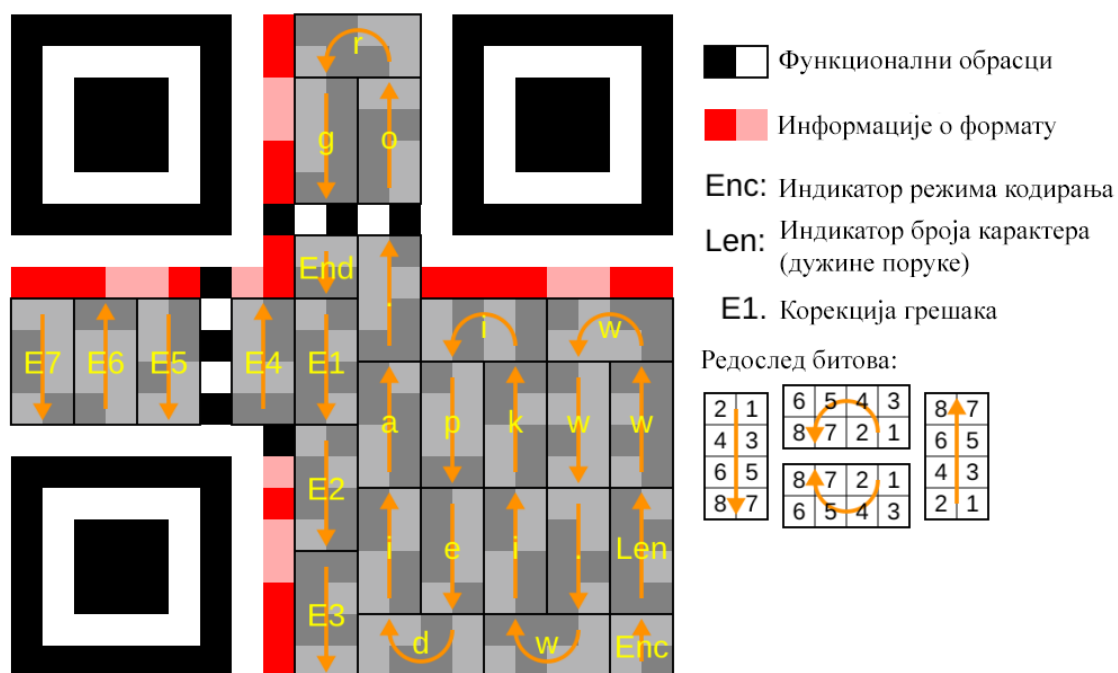
У нашем случају, максимални капацитет је 104 бита, а ми имамо 80. Дакле, фали нам 24 бита, односно 3 цела бајта, која ћемо додати на крај стринга: 00100000 01011011 00001011 01111000 11010001 01110010 11011100 01001101 01000011 01000000 **11101100 00010001 11101100**.

Следећи корак је генерисање кодних речи за корекцију грешака. Ове кодне речи ће се користити као коефицијенти полинома поруке. Нека наш пример користи ниво корекције грешака М, а пошто је верзије 1, креиран код ће бити 1-М. Следеће кодне речи поруке: 00100000 01011011 00001011 01111000 11010001 01110010 11011100 01001101 01000011 01000000 11101100 00010001 11101100 00010001 11101100 00010001, када пребацимо из бинарних у децималне, добијемо следеће бројеве: 32, 91, 11, 120, 209, 114, 220, 77, 67, 64, 236, 17, 236, 17, 236, 17. Ови бројеви су коефицијенти полинома поруке: $32x^{15} + 91x^{14} + 11x^{13} \dots$ Пошто је ово 1-М код, потребно је креирати 10 кодних речи за корекцију грешака. Стога, користићемо следећи полином: $x^{10} + a^{251}x^{67}x^8 + a^{46}x^7 + a^{61}x^6 + a^{118}x^5 + a^{70}x^4 + a^{64}x^3 + a^{94}x^2 + a^{32}x + a^{45}$. Општа формула је:

$$\prod_{i=0}^{d-1} (x + 2^i),$$

где је d број бајтова корекције грешака. Сада је потребно поделити први полином с другим онолико пута колико кодних речи ће бити - у нашем случају, 16 пута. Тиме смо добили полином: $196x^9 + 35x^8 + 39x^7 + 119x^6 + 235x^5 + 215x^4 + 231x^3 + 226x^2 + 93x + 23$. Наших 10 тражених кодних речи за корекцију грешака ће бити коефицијенти добијеног полинома: 196, 35, 39, 119, 235, 215, 231, 226, 93 и 23.

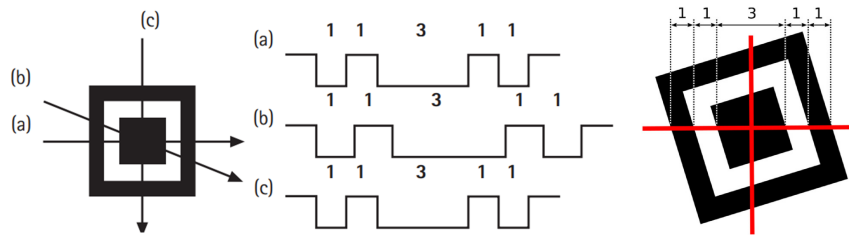
Код већих QR кодова, кодне речи се деле у неколико блокова. Затим се распоређују мешаним редоследом - прве кодне речи из сваког блока редом, па онда друге из сваког блока итд. (слика 2.9). Кодне речи се постављају с десна на лево, почев од доњег-десног ћошка, кривудавим путем, односно „дик-цак”, као на слици 2.8. Након што је цео QR код попуњен, изврши се маскирање и израда QR кода је завршена.



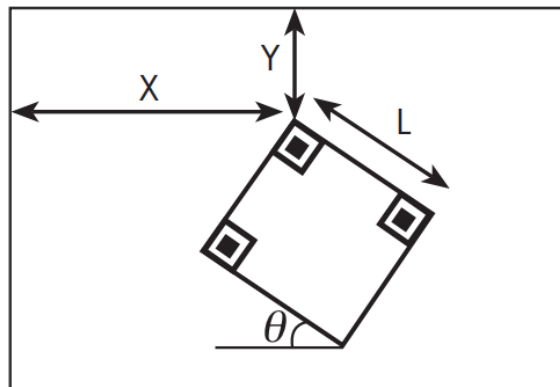
Слика 2.8. Ређање кодних речи у верзији 1 QR кода.

2.2 Скенирање из више углова и исправљање дисторзије

Један од значајних аспеката QR кодова је њихова способност брзог читања из било ког угла. Само читање матричног симбола је имплементирано од стране CCD сензора (енгл. *Charge-Coupled Device*), који се налази у камери. Садржај скенираних линија кода биће потом сачуван у меморији. Потом ће, коришћењем одређеног софтвера, након што су величина и угао позиционе шаре симбола детектовани, процес декодирања бити имплементиран. Под којим год углом да је QR код накривљен, могуће је провући линије кроз центар позиционог квадрата, тако да се тамна и светла поља смењују у односу $1 : 1 : 3 : 1 : 1$ (слика 2.10). Овај однос одабран је након што је Масахирин тим извршио исцрпно истраживање површина слика и симбола одштампаним на флајерима, часописима, картонским кутијама итд, како би дошли до најмање коришћеног односа црних и белих површина на штампаном материјалу. Идентификовањем односа позиција сва три обрасца трагача, могуће је одредити величину (L), угао (θ) и облик симбола истовремено (слика 2.11). Захваљујући упечатљивом позиционом обрасцу три угла, брзина декодирања QR кодова је чак двадесет пута већа него код осталих матричних симбола.

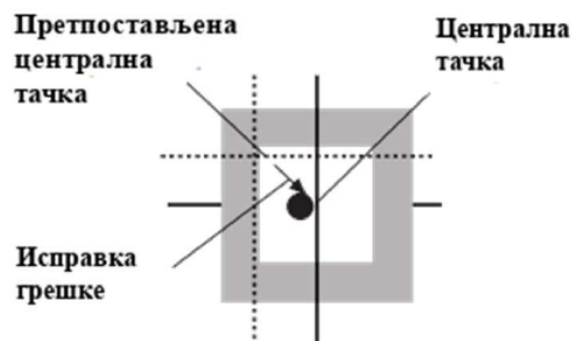


Слика 2.10. Скенирање QR кода из различитих углова.



Слика 2.11. Одређивање угла сликања и величине QR кода.

Исправљање дисторзије QR кода се врши израчунавањем одступања централног положаја обрасца поравнања, процењеног на основу спољашњег облика симбола, од стварног централног положаја обрасца поравнања (слика 2.12).



Слика 2.12. Приказ исправке грешке приликом дисторзије QR кода.

2.3 Корекција грешака

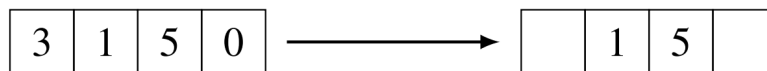
За разлику од других симбола, QR кодови се много чешће могу наћи у неконтролисаним и непредвидивим условима. Врло су могућа оштећења проузрокована лошим временским условима, затим хабањем, гребањем, прљањем или пресавијањем папира; боје могу избледети на сунцу, билборд се може покварити, а QR код залепљен на јавном месту може било ко делимично прекрити стикером, флајером или постером. Ту су и грешке у штампи, лош контраст између позадине и самог кода и лош квалитет камере, као неки од потенцијалних извора деградације података.

Како би се обезбедила поузданост у таквим условима, QR кодови користе софистициран систем за корекцију грешака, заснован на Рид-Соломоновим кодовима (енгл. *Reed–Solomon codes*). Развили су их амерички математичари и инжењери, Ирвинг С. Рид и Густав Соломон 1960. године. Једна од најранијих и најзначајнијих примена била је у мисији Војаџер, која се сматра првом „потпуном” употребом Рид-Соломонових кодова у свемирској мисији. Уз помоћ њих су примљени критични научни подаци (што се наставило до данас), као и прве значајне слике Јупитера и других планета. Користи се такође и при изради дискова, као што су DVD, CD, HDD и Blu-ray. Овај алгоритам омогућава детекцију и исправљање грешака без потребе за поновним слањем или скенирањем кода. Захваљујући овом механизму, QR код може остати функционалан чак и када је до 30% његове површине оштећено или недостаје, у зависности од примењеног нивоа корекције.

2.3.1 Рид-Соломонови кодови

Први случај: грешке услед губитака података на познатим позицијама (енгл. *Erasure errors*)

Нека су информације које желимо да пошаљемо подељене у „пакете”, а неузданост добијених информација потиче из чињенице да су неки пакети изгубљени током преноса, као на слици испод:



Нека се порука коју шаљемо састоји из n пакета и највише k пакета је изгубљено током преноса. Пакети су обележени, тако да знамо тачно који су изгубљени. Дакле, потребно је почетној поруци, која се састоји из n пакета,

додати k пакета вишка, тако да би примаоцу било послато $n + k$ пакета, и он даље може реконструисати оригиналну поруку из било којих n (или више) примљених пакета. Можемо претпоставити, без умањења општости, да је садржај сваког пакета број по модулу q , где је q прост број. На пример, садржај пакета може бити 32-битни стринг и стога му се може доделити вредност између 0 и $2^{32} - 1$; онда бисмо могли изабрати да q буде било који прост број већи од 2^{32} .

Особине полинома над простим пољем $\text{GF}(q)$ (тј. са коефицијентима и вредностима редукованим по модулу q) су савршене за решавање овог проблема и представљају главни део ове шеме за исправљање грешака. Означимо пакете у поруци са $m_1, \dots, m_n$ редом, где је m_i број из $\text{GF}(q)$. Можемо закључити следеће:

1. Постоји јединствени полином $P(x)$ степена $n - 1$, такав да је $P(i) = m_i$ за $1 \leq i \leq n$ (тј. $P(x)$ садржи све информације о поруци, а израчунавање $P(i)$ даје жељени садржај i -тог пакета).

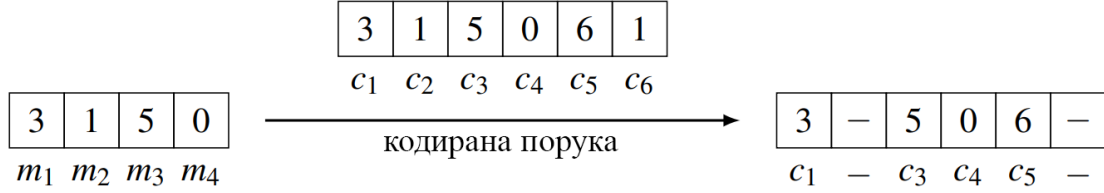
2. Порука је сада $m_i = P(1), \dots, m_n = P(n)$. Можемо генерисати додатне пакете израчунавањем $P(x)$ у тачкама $n + j$ (Подсетимо се да наша послата „кодна реч” (енгл. *codeword*) треба да има вишак информација, тј. треба да садржи више пакета од оригиналне поруке како би се објаснили изгубљени пакети. Ово је разлика између кодне речи и поруке. Кодна реч је оно што се преноси и има редундантности по конструкцији, док је порука нешто што нам корисник даје и не мора нужно да садржи било какав вишак). Дакле, послата кодна реч је $c_1 = P(1), c_2 = P(2), \dots, c_{n+k} = P(n + k)$. Пошто радимо по модулу q , морамо се уверити да је $n + k \leq q$, али овај услов не намеће озбиљно ограничење јер се претпоставља да је q веома велико.

3. Можемо јединствено реконструисати $P(x)$ из његових вредности у било којих n различитих тачака, пошто је степен $n - 1$. То значи да се $P(x)$ може реконструисати из било ког n пренетих пакета (не само оригиналних n пакета). Када реконструишемо полином P , можемо наћи вредност $P(x)$ у тачкама $x = 1, \dots, n$ да бисмо повратили оригиналну поруку $m_1, \dots, m_n$.

Пример

Ана жели да пошаље Марку поруку од $n = 4$ пакета и жели да се заштити од $k = 2$ изгубљена пакета. Затим, под претпоставком да се пакети могу кодирати као цели бројеви између 0 и 6, Ана може да ради над $\text{GF}(7)$ (пошто је $7 \leq n + k = 6$; наравно, у стварним применама, радили бисмо преко много већег поља!). Пошто је $k = 2$, Ана мора да нађе 2 додатне тачке: $P(5) = 6$ и $P(6) = 1$. Сада, Ана може да пренесе кодну реч која се састоји из

$n + k = 6$ пакета, где је $c_j = P(j)$ за $1 \leq j \leq 6$. Дакле, Ана ће послати Марку $c_1 = P(1) = 3, c_2 = P(2) = 1, c_3 = P(3) = 5, c_4 = P(4) = 0, c_5 = P(5) = 6$ и $c_6 = P(6) = 1$. Сада претпоставимо да су пакети 2 и 6 одбачени, у ком случају имамо следећу ситуацију:



Марко је добио вредности 3, 5, 0 и 6. Он користи Лагранжову интерполацију и израчунава следеће полиноме (где све треба интерпретирати мод 7):

$$\begin{aligned} \Delta_1(x) &= \frac{(x-3)(x-4)(x-5)}{(1-3)(1-4)(1-5)} \equiv_7 \frac{(x-3)(x-4)(x-5)}{-24} \equiv_7 \frac{7+1}{4}(x-3)(x-4)(x-5) \\ &\equiv_7 2(x-3)(x-4)(x-5) \end{aligned}$$

$$\Delta_3(x) = \frac{(x-1)(x-4)(x-5)}{(3-1)(3-4)(3-5)} \equiv_7 \frac{(x-1)(x-4)(x-5)}{4} \equiv_7 2(x-1)(x-4)(x-5)$$

$$\begin{aligned} \Delta_4(x) &= \frac{(x-1)(x-3)(x-5)}{(4-1)(4-3)(4-5)} \equiv_7 \frac{(x-1)(x-3)(x-5)}{-3} \equiv_7 \frac{1}{7-3}(x-1)(x-3)(x-5) \\ &\equiv_7 2(x-1)(x-3)(x-5) \end{aligned}$$

$$\begin{aligned} \Delta_5(x) &= \frac{(x-1)(x-3)(x-4)}{(5-1)(5-3)(5-4)} \equiv_7 \frac{(x-1)(x-3)(x-4)}{8} \equiv_7 \frac{1}{7+1}(x-1)(x-4)(x-5) \\ &\equiv_7 (x-1)(x-3)(x-4) \end{aligned}$$

Затим, реконструише полином $P(x)$:

$$\begin{aligned} P(x) &= 3\Delta_1(x) + 5\Delta_3(x) + 0\Delta_4(x) + 6\Delta_5(x) \\ &= 6(x-3)(x-4)(x-5) + 10(x-1)(x-4)(x-5) + 6(x-1)(x-3)(x-4) \\ &= 22x^3 - 220x^2 + 686x - 632 \\ &= (7 \cdot 3 + 1)x^3 + (7 \cdot (-32) + 4)x^2 + (7 \cdot 98)x - 7 \cdot 91 + 5 \equiv x^3 + 4x^2 + 5 \pmod{7} \end{aligned}$$

Марко затим може да израчуна $m_2 = P(2) = 29 \equiv 1 \pmod{7}$, што је пакет који је изгубљен из оригиналне кодне речи. Генерално, без обзира која два пакета су одбачена, пратећи потпуно исту методу, Марко увек може реконструисати $P(x)$ и самим тим оригиналну поруку.

Размотримо сада шта би се десило ако би Ана послала један пакет мање. Ако би послала c_j само за $1 \leq j \leq n+k-1$, онда би са k брисања, Марко примио c_j само за $n-1$ различитих вредности j . Дакле, он не би могао да реконструира $P(x)$ (пошто постоји q полинома степена највише $n-1$ који се слажу са $n-1$ пакетима које је Марко примио). Ова шема за исправљање грешака је стога оптимална: може да поврати н карактера пренете поруке помоћу било којих н примљених карактера, али опоравак помоћу било ког мањег броја карактера је немогућ.

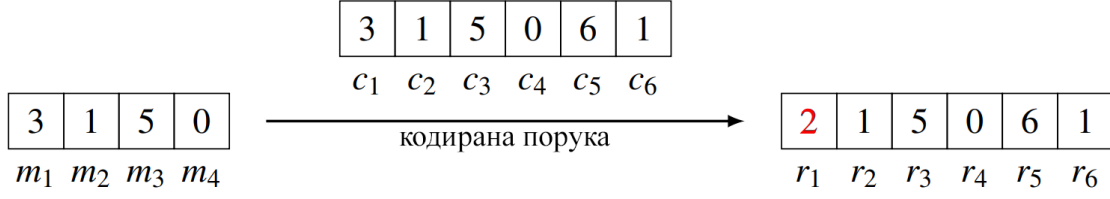
Други случај: грешке услед корупције података на непознатим позицијама)

Сада, размотримо много изазовнији сценарио. Претпоставимо да Ана жели да комуницира са Марком преко канала са шумом (рецимо, путем модема). Њена порука је $m_1, \dots, m_n$, где можемо сматрати m_i карактерима (било бајтовима или словима). Проблем је сада што су неки од карактера оштећени током преноса због шума канала. Тако Марко прима тачно онолико знакова колико Ана пренесе, али k од њих је оштећено, и Марко нема појма којих k карактера је у питању. Опоравак од таквих општих грешака је много изазовнији од опоравка од грешака брисања, али Ана се и даље може заштитити од k општих грешака, на рачун преношења само $2k$ додатних знакова (дупло више него у случају брисања у првом случају).

Поново ће сваком карактеру бити додељен број по модулу q за неки прости број q . (За енглеску абецеду, q је неки прости број већи од 26, рецимо $q = 29$.) Као и раније, поруци можемо доделити полином $P(x)$ степена $n-1$ над $\mathbb{GF}(q)$, тако да је $P(1) = m_1, \dots, P(n) = m_n$. Да би се заштитила од k општих грешака, Ана мора пренети $2k$ додатних знакова: стога је кодна реч $c_1, \dots, c_{n+2k}$ где је $c_j = P(j)$ за $1 \leq j \leq n+2k$, и $n+k$ ових карактера које Марко прими су неоштећени. Такође, мора важити $q > n+2k$. На пример, ако Ана жели да пошаље $n = 4$ карактера Марку преко модема у коме је $k = 1$ од карактера оштећен, она мора редундантно послати кодирану поруку која се састоји од 6 карактера. Претпоставимо да жели да пренесе исту поруку (3150) као у нашем примеру брисања у првом случају, и да је c_1 оштећен са 3 на 2.

Овај сценарио је приказан на следећој слици:

Са Марковог становишта, проблем реконструкције Анине поруке је исти као и реконструкција полинома $P(x)$ из $n+2k$ примљених карактера $r_1, r_2, \dots, r_{n+2k}$. Другим речима, Марку је дато $n+2k$ вредности, $r_1, r_2, \dots, r_{n+2k}$ по модулу



q , са обећањем да постоји полином $P(x)$ степена $n - 1$ над $\text{GF}(q)$ такав да је $P(i) = r_i$ за $n + k$ различитих вредности i између 1 и $n + 2k$. Марко мора реконструисати $P(x)$ из ових података (у горњем примеру, $n + k = 5$, и $r_2 = P(2) = 1$, $r_3 = P(3) = 5$, $r_4 = P(4) = 0$, $r_5 = P(5) = 6$, $r_6 = P(6) = 1$). Међутим, он не зна које од $n + k$ вредности су тачне.

Да ли Марко уопште може да реконструише $P(x)$? Претпоставимо да је $P(i) = r_i$ за $n + k$ тачака, и $P'(i) = r_i$ за $n + k$ (можда различитих) тачака. Пошто укупно постоји само $n + 2k$ тачака, ови скупови морају да се преклапају на n тачака ($2(n + k) - (n + 2k) = n$), и стога је $P(i) = r_i = P'(i)$ за n тачака, што имплицира да морају бити полиноми истог степена $n - 1$. Штавише, можемо приметити да за оригинални полином важи $P(i) = r_i$ за $n + k$ тачака јер постоји највише k грешака. Дакле, $P(x)$ је једини полином који задовољава дате услове.

Марков задатак је да пронађе полином $P(x)$ степена $n - 1$ такав да је $P(i) = r_i$ за најмање $n + k$ вредности i . Ако то може да уради, може бити сигуран да је полином који је пронашао заправо оригинални полином $P(x)$. Нека су $e_1, \dots, e_k$ локације на којима су се појавиле грешке (тако да је $P(i) = r_i$ за све $i \in e_1, \dots, e_k$). Марко не зна где су ове грешке. Како не би морао да погађа где се оне налазе, искористиће следећи полином k -тог степена, такозвани „полином локатор грешака” (енгл. *error-locator polynomial*):

$$E(x) = (x - e_1)(x - e_2)(x - e_k).$$

За овај полином важе следеће важне једнакости:

$$P(i)E(i) = r_i E(i) \text{ за } 1 \leq i \leq n + 2k. \quad (2.1)$$

Једнакости важе у тачкама у којима није дошло до грешке јер је у тим тачкама $P(i) = r_i$, али и тачкама у којима је дошло до грешке јер је тада $E(i) = 0$. Ово запажање чини основу Берлекемп-Велч алгоритма декодирања (*Berlekamp-Welch algorithm*). Развили су га амерички математичари Елвин Р. Берлекемп и Лојд Р. Велч 1986. године. Из једнакости у (2.1), може се добити $n + 2k$ линеарних једначина са $n + 2k$ непознатих, из којих се локације грешака и коефицијенти $P(x)$ могу лако извести (аналогно интерполацијској шеми из првог случаја).

Дефинишимо полином $Q(x) := P(x)E(x)$, који је степена $n + k - 1$ и има коефицијенте $a_0, a_1, \dots, a_{n+k-1}$. Полином $E(x) = (x - e_1)(x - e_k)$ има $k + 1$ коефицијената $b_0, b_1, \dots, b_k$, али је водећи коефицијент (b_k) увек 1. Дакле, можемо написати:

$$Q(x) = a_{n+k-1}x^{n+k-1} + a_{n+k-2}x^{n+k-2} + \dots + a_1x + a_0;$$

$$E(x) = x^k + b_{k-1}x^{k-1} + \dots + b_1x + b_0.$$

Када заменимо $Q(x) = P(x)E(x)$ у (2.1), добијемо $n + 2k$ једнакости $Q(i) = r_i E(i)$, за $1 \leq i \leq n + 2k$. Записивањем i -ту једначину користећи коефицијенте од $Q(x)$ и $E(x)$, добијамо

$$a_{n+k-1}i^{n+k-1} + a_{n+k-2}i^{n+k-2} + \dots + a_1i + a_0 \equiv r_i(i^k + b_{k-1}i^{k-1} + \dots + b_1i + b_0) \pmod{q}$$

Ово је систем од $n + 2k$ линеарних једначина, по једна за сваку вредност i , са $n + 2k$ непознатих $a_0, a_1, \dots, a_{n+k-1}, b_0, b_1, \dots, b_{k-1}$. Решавањем овог система, добијамо $E(x)$ и $Q(x)$. Затим можемо израчунати $\frac{Q(x)}{E(x)}$ да бисмо добили $P(x)$.

Пример

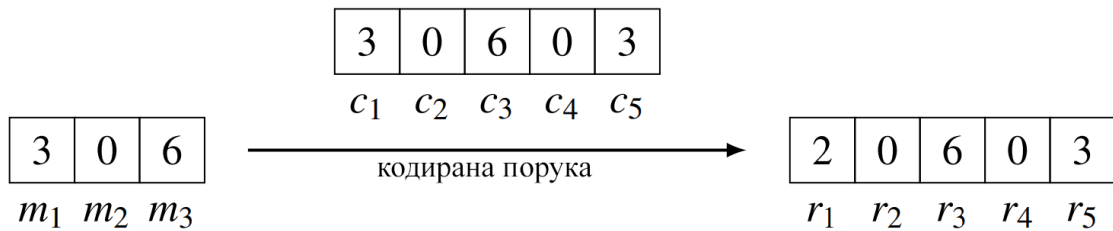
Претпоставимо да радимо над $\text{GF}(7)$ и Ана жели да пошаље Марку $n = 3$ карактера „3”, „0” и „6” преко модема. Ово је еквивалентно нпр. коришћењу само првих 7 слова абетеде, где је $a = 0, \dots, g = 6$. Дакле, порука коју Ана жели да Марко прими је „dag”. Она ће затим интерполацијом (нпр. Лагранжовом) добити полином

$$P(x) = x^2 + x + 1,$$

што је јединствени полином степена 2 такав да је $P(1) = 3$, $P(2) = 0$ и $P(3) = 6$.

Нека Ана зна да до $k = 1$ карактер може бити оштећен; она онда треба да пошаље $n + 2k = 5$ карактера $P(1) = 3$, $P(2) = 0$, $P(3) = 6$, $P(4) = 0$ и $P(5) = 3$ Марку. Претпоставимо да је $P(1)$ заправо оштећен и Марко прима карактер 2 уместо 3 (тј. Ана шаље кодну реч „dagad”, али Марко уместо тога прима „sagad”), као на слици:

Нека је $E(x) = (x - e_1) = x + b_0$ и $Q(x) = a_3x^3 + a_2x^2 + a_1x + a_0$. Марко ће убацити



вредности $i = 1, i = 2, \dots, i = 5$ у једначине 2.1. Сређивањем (рачунајући по модулу 7) ће добити следећих 5 једначина с 5 непознатих:

$$\begin{aligned}a_3 + a_2 + a_1 + a_0 + 5b_0 &= 2 \\a_3 + 4a_2 + 2a_1 + a_0 &= 0 \\6a_3 + 2a_2 + 3a_1 + a_0 + b_0 &= 4 \\a_3 + 2a_2 + 4a_1 + a_0 &= 0 \\6a_3 + 4a_2 + 5a_1 + a_0 + 4b_0 &= 1\end{aligned}$$

Решавањем система, добија $a_3 \equiv 1, a_2 \equiv 0, a_1 \equiv 0, a_0 \equiv 6$ и $b_0 \equiv 6 \pmod{7}$. Сада зна полиноме $Q(x) = x^3 + 6 \equiv_7 x^3 + 1$ и $E(x) = x + 6 \equiv_7 x - 1$. Такође закључује да је до грешке дошло на позицији $e_1 = 1$, односно да је први карактер корумпиран. Затим срачуна $P(x) = \frac{Q(x)}{E(x)} = \frac{x^3-1}{x-1} = x^2 + x + 1$. Израчунавањем $P(1) = 3 = „d”$, Марко добија оригиналну, неоштећену поруку „dag”.

Синдромско декодирање

Иако се у пракси за декодирање QR кодова не користи Берлекемп-Велч алгоритам, овај метод представља једноставан приступ за разумевање Рид-Соломонових кодова и основних принципа детекције и корекције грешака, због своје јасне математичке структуре и интуитивне идеје интерполације. Берлекемп-Велч алгоритам је представљен као теоријски оквир који осветљава основне идеје које стоје иза Рид-Соломоновог декодирања.

У стварним имплементацијама QR кодова користи се другачији приступ — **синдромско** декодирање, најчешће путем Берлекемп-Мејси алгоритма (енгл. *Berlekamp-Massey algorithm*). Овај алгоритам је ефикаснији у погледу меморије и брзине (временске је сложености $O(n^2)$, док Берлекемп-Велч алгоритам има сложеност $O(n^3)$), што га чини погодним за уређаје са ограниченим ресурсима, као што су мобилни скенери QR кодова. Уместо интерполације, Берлекемп-Мејси користи унапред израчунате синдроме да би ефикасно пронашао полином локатор грешака и извршио корекцију.

Синдроми се рачунају на основу примљених података и представљају резултат евалуације полинома у одређеним тачкама. Формула за синдром је:

$$S(x) = \sum_{i=0}^{k-1} r(2^i) \cdot x^i$$

Ако је вредност свих синдрома нула, нема грешке. У супротном, користи се Берлекемп-Мејси алгоритам како би се пронашао полином локатор грешака, који открива где се налазе грешке у подацима. Након тога се користи Chien

претрага за лоцирање тачних позиција грешака у низу података. На крају, уз помоћ Forney алгоритма, израчунава се и вредност грешака, које се затим одузимају од примљених података како би се добили тачни, оригинални подаци. Рид-Соломонов код може исправити двоструко више брисања него грешака, а било која комбинација грешака и брисања може се исправити све док је задовољена релација $2E + S \leq n - k$, где је E број грешака, а S број брисања у блоку.

Требало би напоменути да се операције врше над пољем $GF(2^8)$ - 256 није прост број, али је значајан јер су тиме резултати свих математичких операција бројеви који се могу представити као осмобитни бајтови. Подаци се шаљу као кодирани блокови, а број симбола у кодираном блоку је $n = 2^m - 1$. Дакле, Рид-Соломонов код који ради са 8-битним симболима има $n = 2^8 - 1 = 255$ симбола по блоку. Код већих QR код симбола, порука се дели на неколико блокова Рид-Соломон кода. Величина блока је изабрана тако да се не покушава исправити више од 15 грешака по блоку; ово ограничава сложеност алгоритма за декодирање. Блокови кода се затим испреплићу, што смањује вероватноћу да ће локализовано оштећење QR код симбола преоптеретити капацитет било ког појединачног блока. На пример, QR код верзије 1 са корекцијом грешака нивоа L састоји се од једног блока за корекцију грешака са укупно 26 бајтова кода (направљених од 19 бајтова поруке и седам бајтова за корекцију грешака). Он може да исправи грешке до 2 бајта. Стога је овај код познат као код за корекцију грешака (26,19,2) над $GF(2^8)$. Понекад се представља и скраћено, као код (26,19).

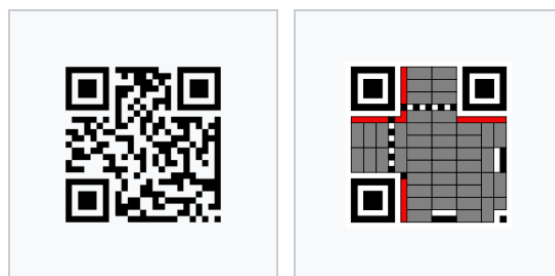
Особине Рид-Соломонових кодова чине их посебно погодним за примене где се грешке јављају у групама, што је чест случај за QR кодове. То је зато што коду није битно колико је битова у симболу погрешно — ако је више битова у симболу оштећено, то се рачуна само као једна грешка.

Коришћење „природне” величине блокова Рид-Соломон кода није обавезно. Техника позната као „скраћивање” може произвести мањи код било које жељене величине из већег кода. На пример, широко коришћени код (255,223) може се конвертовати у код (160,128) попуњавањем неискоришћеног дела изворног блока са 95 бинарних нула и њиховим непреношењем. На декодеру, исти део блока се локално учитава бинарним нулама. Тако нпр. QR код верзије 3 (29×29) има поруку од 26 бајтова података, кодирану помоћу два блока Рид-Соломон кода. Сваки блок је Рид-Соломон код (255, 233) скраћен на код (35, 13).

3 Додатни аспекти QR кода

3.1 Верзије QR кода

Модел 1 је стара верзија QR кода. Разликује се по томе што нема образац поравнања, а има додатне позиционе обрасце. Модерни QR код се сматра моделом 2.



Слика 3.1. Модел 1.

Микро QR код је мања верзија QR кода, која се користи када је простор за симбол ограничен. Може да садржи највише 35 цифара, 21 ASCII карактера или 15 бајтова.



Слика 3.2. Микро QR код.

Правоугаони микро QR код (енгл. *rMQR Code*) је погоднији за правоугаоне површине. Развијен је 2022. године.



Слика 3.3. Правоугаони микро QR код.

iQR код је алтернатива за QR код. Може бити и квадратног и правоугаоног облика. Њему иста количина информација заузима 30% мање места, и има још једну опцију нивоа корекције грешака, ниво S, који омогућава корекцију и до 50% оштећеног кода.



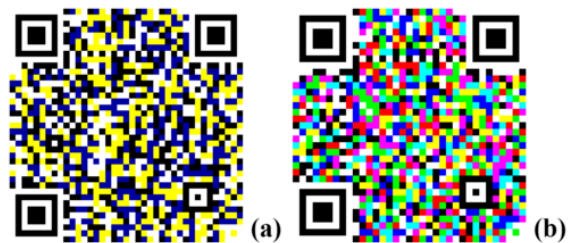
Слика 3.4. iQR код.

Frame QR код подсећа на рам за слику. У његову унутрашњост се може ставити било који дизајн.



Слика 3.5. Frame QR код.

High Capacity Colored 2-Dimensional (HCC2D) Code има модуле различитих боја.

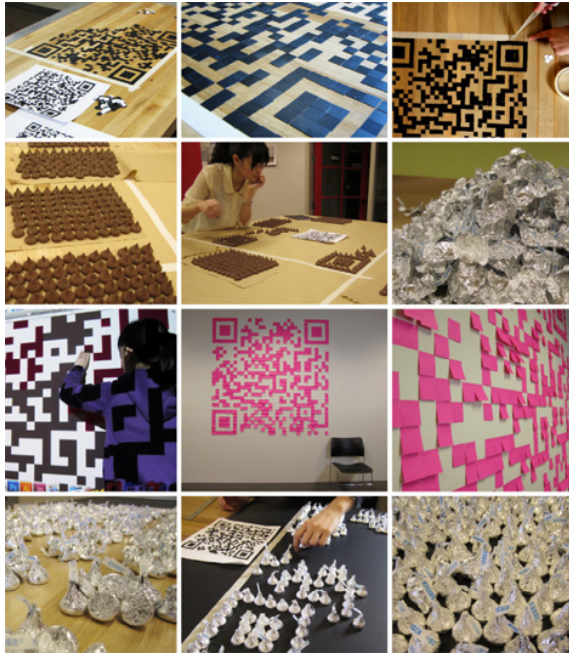


Слика 3.6. HCC2D код.

3.2 Занимљиве примене

У овом поглављу представљене су само неке од креативних и оригиналних употреба QR кодова, које илуструју како је ова технологија пронашла место у неочекиваним контекстима – од маркетинга до уметничких дела.

The Curiosity Project



У оквиру овог пројекта, дизајнерке Lauren Manning и Camile Wei-Hsin Lin су од свакодневних предмета ручно направиле серију QR кодова, који су водили к веб-сајту The Curiosity Project блога. Сlike њих биле су постављене на билбордима широм Њујорка. Идеја пројекта је била да подстакне људе да проналазе инспирацију свуда око себе.



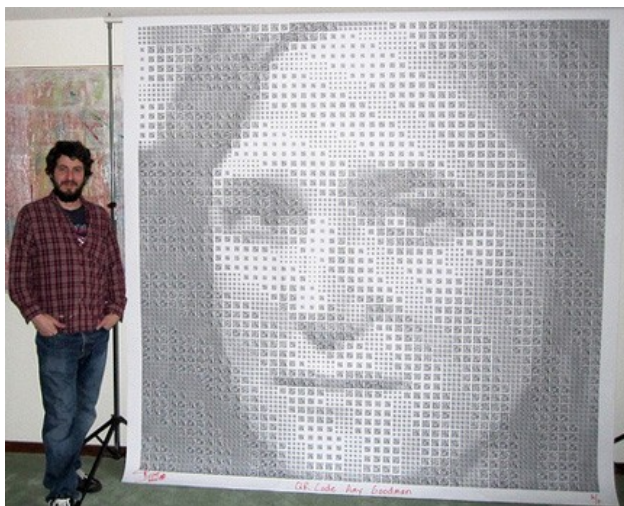
Слика 3.7. QR код на првој слици је нацртан маркером; на другој је направљен од Post-it лепљивих папирића, а на трећој од магнета.



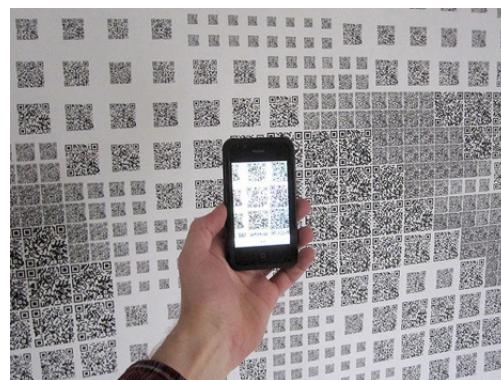
Слика 3.8. QR кодови направљени од Hershey's Kisses чоколадица: на левој слици су с омотом, а на десној без.

Портрет од QR кодова

Уметник Скот Блејк направио је портрет новинарке Ејми Гудмен користећи чак 2304 QR кодова. Ови QR кодови воде до снимака емисије, током периода од 9 година, у којој је она била водитељка.



Слика 3.9. Портрет издалека.



Слика 3.10. Портрет изблиза.

Модерна библиотека



Слика 3.11. Хибридна библиотека на станици метроа.

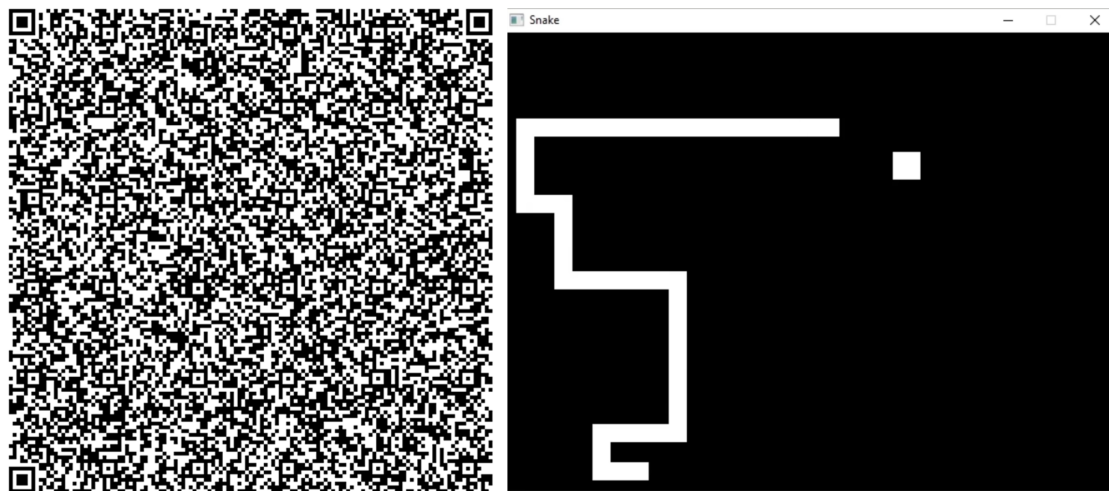
На зидовима станице метроа у Букурешту, од пода до плафона, протеже се постер са сликама корица књига (слика ??). Свака „књига” има QR код на себи, који води ка њеном одломку, како би путници имали штиво за читање док чекају превоз.

По истом принципу функционише куповина намирница на неким станицама јавног превоза у Јужној Кореји. На зидовима су одштампане полице с производима обележеним QR кодовима, чијим скенирањем они бивају додати у дигиталну корпу.



Игрица

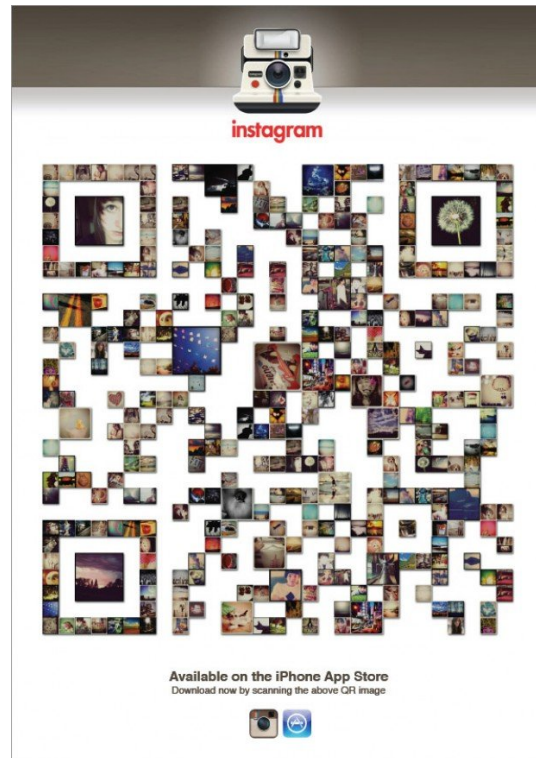
Програмер и јутјубер MattKC је направио QR код који садржи бинарни код за познату игрицу „Змија” (енгл. *Snake Game*). Користио је верзију 40 QR кода и игрицу је могуће покренути директно као EXE фајл на компјутеру скенирањем веб-камером. Процес прављења овог QR кода налази се на његовом Јутјуб каналу.



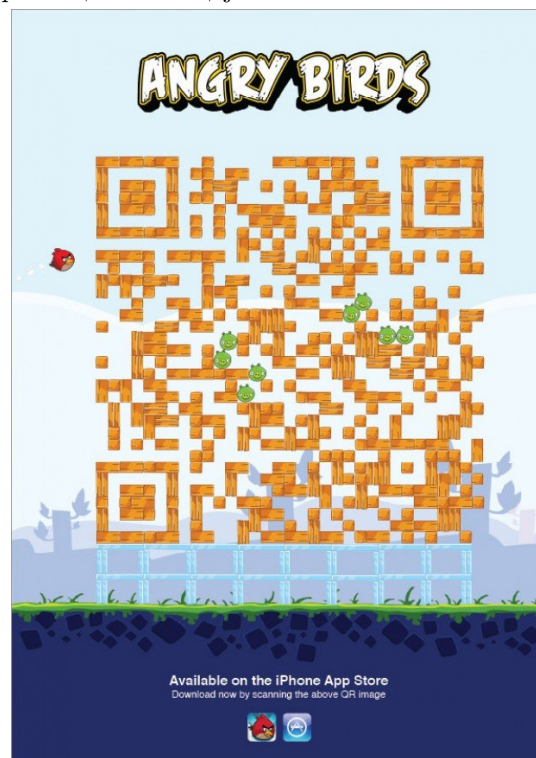
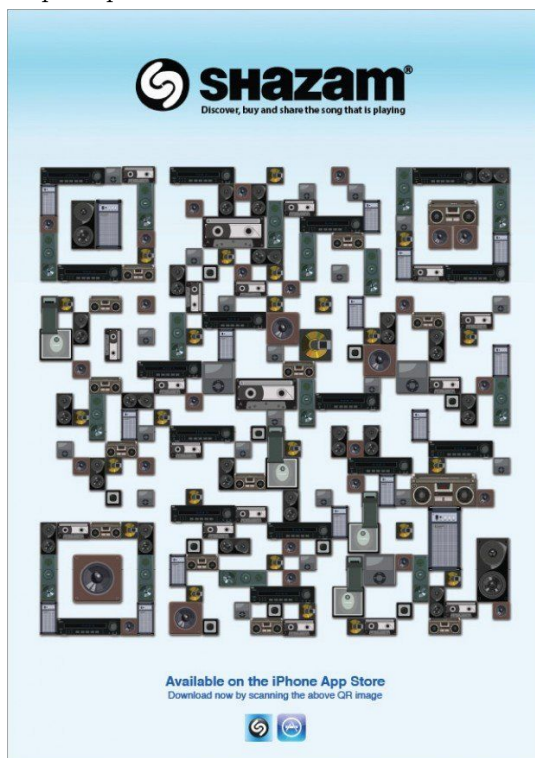
Слика 3.12. QR код и покренута игрица „Змија”.

Креативне рекламе и маркетиншке кампање

Компаније за развој мобилних апликација понекад користе креативно дизајниране QR кодове као део маркетиншких кампања, комбинујући функционалност са визуелном привлачношћу. Овине QR код не служи само за брзо преусмеравање к апликацији, већ постаје и упечатљив визуелни елемент који привлачи пажњу и подстиче интеракцију. Овакав приступ повећава вероватноћу да корисници заиста скенирају код, јер их истовремено информисе и забавља. Неки примери ове врсте маркетинга приказани су на следећим сликама:

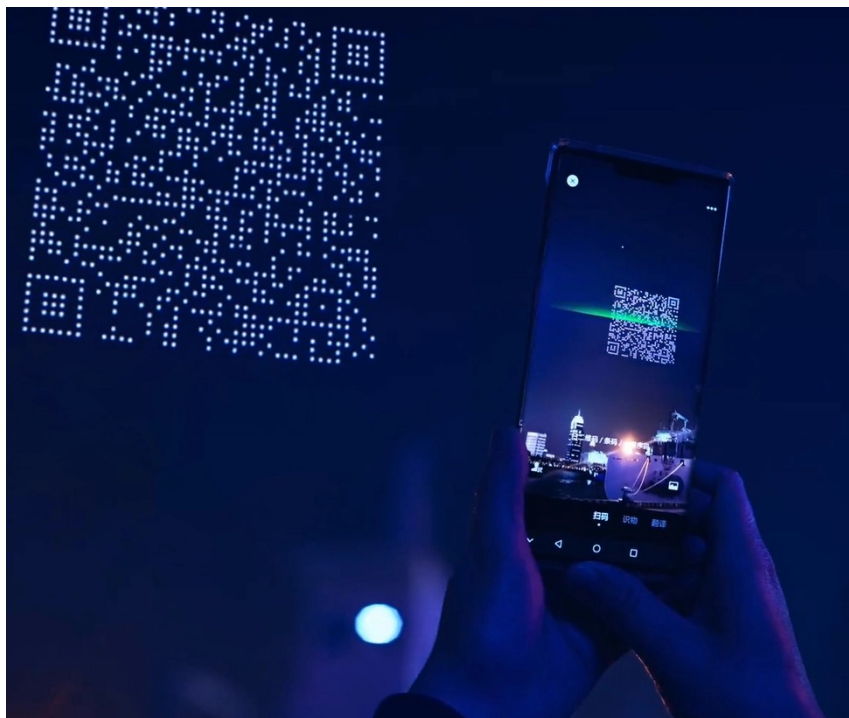


(а) QR код направљен да личи на хамбургер (б) Реклама за друштвену мрежу Инста- као реклама за апликацију која нуди купоне грам. QR код се састоји из слика које су ко- за ресторане. рисници апликације постављали.



(в) Реклама за Shazam, апликацију за пре- (г) QR код с елементима игрице Angry Birds, познавање песама. QR код је направљен од као реклама за пју. аудио уређаја.

Кинеске компаније Cugames и Bili Bili су 2011. године направиле промоцију за нову игрицу користећи 1500 светлећих дрона на ноћном небу. Дронови су формирали облике неких ликова из игрице, а на крају програма QR код (слика 3.14) помоћу ког се може скинути игрица.



Слика 3.14. дрони формирају QR код.

Године 2011, SF Bio, један од највећих шведских ланаца биоскопа, организовао је „потрагу за благом”. Као део кампање, постављени су билборди са текстом направљеним од 687 QR кода. Оглас је подстакао људе да пронађу прави QR код који би им пружио прилику да бесплатно гледају филмове током целог лета у биоскопима овог ланца.



3.3 Безбедност

QR кодови су изузетно корисни, али њихова употреба нажалост носи и одређене безбедносне ризике. Један од главних проблема је то што корисници не могу видети шта се заиста крије иза QR кода пре него што га скенирају. Злонамерни актери могу направити лажне QR кодове који воде ка фишинг сајтовима или садрже малициозне линкове. Скенирањем таквог кода, корисник може несвесно преузети малвер на свој уређај. Тако је 2011. године Лабораторија Касперски уочила први злонамерни QR код. Након скенирања кода, корисник би био пребачен на веб-сајт са ког би се фајл са злонамерним софтвером инсталирао без знања корисника. Коришћење QR кодова за фишинг је корисно јер већина људи сада сумња у линкове у мејловима и (с правом) су опрезни када је у питању кликтање на скраћене линкове. Криминалци стога користе QR кодове како би прикрили линкове ка злонамерним веб-сајтовима које садрже фишинг имејлови. Такође, нису сви безбедносни алати за откривање фишинг мејлова дизајнирани за скенирање слика, тако да QR код може лакше да се провуче.

QR кодови могу бити коришћени и за преусмеравање на сајтове који краду личне податке или податке о платним картицама. Постоји и ризик да физички QR код буде замењен злонамерним, на пример, залепљен преко оригиналног кода на јавним местима. У неким случајевима, QR код може аутоматски покренути акције као што су слање порука или позивање бројева без знања корисника. Посебан проблем представљају апликације за скенирање које саме могу бити небезбедне. Због тога је важно користити само проверене апликације, бити опрезан приликом скенирања кодова са непознатих извора и на јавном месту проверити да QR код није физички измењен на било који начин.

4 Закључак

Кроз овај рад показано је да су QR кодови много више од обичних црно-белих квадрата које срећемо на амбалажи производа, флајерима или менијима у ресторанима. Они представљају резултат деценијског технолошког развоја, у коме се преплићу инжењеринг, математика и потребе савременог човека за брзом и поузданом комуникацијом.

Њихова архитектура омогућава не само ефикасно чување и пренос велике количине података, већ и отпорност на грешке, флексибилност у примени и лакоћу читања из различитих углова. Захваљујући отвореном формату и једносутавној употреби, QR кодови улазе у све сфере живота: од индустрије и медицине до уметности, образовања и свакодневне комуникације. Њихова примена наставља да се шири, а иновације попут динамичких QR кодова показују да се ова технологија тек развија.

Надам се да ће овај рад подстаћи читаоце да следећи пут, када угледају QR код, не помисле само на брз начин да отворе неки линк, већ да препознају у њему читаву причу о људској креативности, иновацијама и потреби да информација буде доступна свима – брзо, ефикасно и безбедно.

Литература

- [1] <https://www.youtube.com/watch?v=w5ebcowAJD8>,
датум приступа: 11. 11. 2024.
- [2] <https://rg.edu.rs/wp-content/uploads/2022/01/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B8%D1%80%D0%B0%D1%9A%D0%B5-%D0%B8-%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%81%D0%BA%D0%B8-%D1%98%D0%B5%D0%B7%D0%B8%D1%86%D0%B8-%D0%82%D0%BE%D1%80%D1%92%D0%B5-%D0%A8%D1%83%D0%BB%D0%B5%D0%B8%D1%9B.pdf>,
датум приступа: 1. 5. 2024.
- [3] <https://www.britannica.com/biography/Samuel-F-B-Morse>,
датум приступа: 3. 5. 2024.
- [4] <https://www.mentalfloss.com/posts/what-hath-god-wrought-meaning#:~:text=Annie%20chose%20the%20quote%20%E2%80%9CWhat,humankind%20a%20crucial%20new%20tool.>,
датум приступа: 3. 5. 2024.
- [5] <https://www.history.com/this-day-in-history/may-24/what-hath-god-wrought>,
датум приступа: 3. 5. 2024.
- [6] John Joseph Fahie, „A history of electric telegraphy, to the year 1837”, 1884.
- [7] Tony Seideman, „Barcodes Sweep the World”, чланак из часописа Wonders of Modern Technology, пролеће 1993. год.
- [8] <https://www.idsapp.com/how-do-barcodes-work/?amp>,
датум приступа: 8. 5. 2024.

- [9] <https://da-proj.gitlab-pages.liu.se/vanheden/pdf/ean13.pdf>,
датум приступа: 13. 5. 2024.
- [10] <https://expresscorp.com/wp-content/uploads/2023/02/USS-49.pdf>,
датум приступа: 15. 5. 2024.
- [11] <https://barcodeguide.seagullscientific.com/content/Symbologies/PDF417.htm>,
датум приступа: 16. 5. 2024.
- [12] <https://web.archive.org/web/20120524085651/http://accusoft.com/whitepapers/barcodes/BarcodesinDocuments-BestPractices.pdf>,
датум приступа: 16. 5. 2024.
- [13] <https://mainiichi.jp/english/articles/20211109/%a/00m/0by/024000c>,
датум приступа: 17. 5. 2024.
- [14] https://d1wqtxts1xzle7.cloudfront.net/51791265/Three_QR_Code-libre.pdf?1487081486=&response-content-disposition=inline%3B+filename%3DThree_QR_Code.pdf&Expires=1748005769&Signature=STmCyD84QvgaYVfwbwie6yBSA49UC8W5ftJEAi69GDC0tMYnks3l8vuc7gz2le1E6b3P-QroY_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA,
датум приступа: 19. 5. 2024.
- [15] <https://www.thonky.com/qr-code-tutorial/>,
датум приступа: 20. 5. 2024.
- [16] https://d1wqtxts1xzle7.cloudfront.net/51791265/Three_QR_Code-libre.pdf?1487081486=&response-content-disposition=inline%3B+filename%3DThree_QR_Code.pdf&Expires=1748559766&Signature=REGuJX6Yw56bjrbu1M8Od6SAwrcitjzDicsT9FGP8zkQM5tBB3Cla2ZE22NBPbJFyjgj62h5i0_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA,
датум приступа: 20. 5. 2024.
- [17] <https://www.youtube.com/watch?v=1pQJkt7-R4Q&t=672s>,
датум приступа: 20. 5. 2024.
- [18] https://git.kb28.ch/HEL/5D_Heredero_Louis_TM2022/raw/commit/ea1fbe4e9494c58e48bdfaa559e5592870f65e38/latex/main.pdf,
датум приступа: 21. 5. 2024.

-
- [19] <https://www.eecs70.org/assets/pdf/notes/n9.pdf>,
датум приступа: 23. 5. 2024.
- [20] <https://cargocollective.com/thedesignzoo/The-Curiosity-Project>,
датум приступа: 24. 5. 2024.
- [21] <https://weburbanist.com/2011/05/26/hidden-creativity-qr-code-portrait-hides-double-meaning/>,
датум приступа: 24. 5. 2024.
- [22] <https://weburbanist.com/2013/02/09/hybrid-library-qr-codes-access-ebooks-in-subway-station/>,
датум приступа: 25. 5. 2024.
- [23] <https://www.youtube.com/watch?v=ExwqNreocpg>,
датум приступа: 25. 5. 2024.