ELEKTROTEHNIŠKI VESTNIK 89(5): 233–238, 2022
IZVIRNI STROKOVNI ČLANEK

Temperaturna sledljivost transporta izdelkov z uporabo termokromnih pigmentov v pametni QR-kodi

Rok Hrovat, Aleksander Sešek

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška cesta 25, 1000 Ljubljana, Slovenija
E-pošta: rh8968@student.uni-lj.si, aleksander.sesek@fe.uni-lj.si

Povzetek. V članku sta predstavljeni QR-koda in njena zmožnost, da ohrani zapisane informacije tudi, če je del kode poškodovan, oziroma v našem primeru nadomeščen s funkcionaliziranim področjem. To področje lahko vsebuje različne vrste podatkov, ki so prav tako kodirani. Ker so QR-kode vse pogosteje uporabljene kot sodobna oblika označevanja izdelkov, spletnih povezav, menijev ipd., lahko ta dodaten podatek vsebuje pomembne dinamične informacije o izdelku. V članku je opisana ena od možnosti, ki omogoča nadzor temperaturnega profila okolja, po katerem se neki izdelek giblje. To informacijo se vključi v pametno QR-kodo. Ena izmed rešitev za temperaturno sledenje so temperaturno odvisni pigmenti, ki so lahko izdelani za različna temperaturna območja. Predstavljena je QR-koda z dodanim sredinskim barvnim območjem, ustrezen čitalnik za tako kodo in prenos informacij na strežnik za morebitno nadaljnjo uporabo.

Ključne besede: QR-koda, čitalnik, ESP32-CAM, barvno kodiranje, termokromizem

Tracking the product temperature in transportation using thermochromatic pigments in a smart QR Code

The paper presents a QR code and its ability to maintain the information even when part of the code is damaged, or, in our case, replaced with a functionalized field, which carries different types of information which are additionaly coded. As the QR code is used for advanced product labels, links, menus, etc., an additional information provides an important dynamic information about the product. The main issue solved in the paper is how to keep the temperature track of the product environment and incorporate it in the QR code. The solution for temperature tracking are temperature irreversible pigments functionalized for different temperature ranges. The paper presents the QR code with a functionalized central area, reader of such code and information transfer to the server for future processing.

1 UVOD

V industriji je nadzor vseh vrst izdelkov zelo pomemben, še posebej pri množični proizvodnji. Zato je bilo razvitih več nadzornih sistemov, od katerih je črtna koda ena izmed najbolj razširjenih. Srečamo jo tako rekoč na vsakem izdelku, ki je danes proizveden. Črtna koda je enostavna rešitev za pridobivanje informacij o izdelku, vendar ne more podajati dinamičnih informacij, kot so na primer temperatura skladiščenja, vlaga, osvetljenost itd., predvsem če gre za pokvarljive izdelke v prehranski industriji.

Na kakovost izdelkov vpliva mnogo različnih dejavnikov, kot so vlaga, osvetlitev, mehanska obremenitev, temperatura in podobno. Pri transportu izdelkov je treba zagotoviti ustrezne pogoje za ohranitev njegove neoporečnosti. Eno od največjih težav pri tem predstavlja zagotavljanje temperaturne sledljivosti gibanja izdelka, saj ta ne sme preseči določenega temperaturnega območja. Tipičen primer je odmrznitev zamrznjenih živil in na primer pregretje ali zamrznitev zelenjave med transportom. Nedopustno je, da bi se globoko zamrznjeno živilo med transportom odtalilo, pozneje pa bi bilo zopet zamrznjeno in prodano. Rešitev je v sledljivosti temperature živila v prodajni verigi, ki je vidna na embalaži izdelka. Ideja je nadomestitev klasične črtne kode s QR-kodo, ki poleg osnovnih podatkov o produktu (kot so vrsta izdelka, datum in kraj proizvodnje ipd.) vsebuje tudi dodatno informacijo o temperaturnem področju gibanja izdelka v obliki barve. S tem zagotovimo sledljivost in hkrati nadzor nad temperaturnim profilom okolja, po katerem se je izdelek gibal. V članku predstavljeno rešitev lahko uporabimo pri vseh izdelkih, občutljivih za temperaturne variacije.

Prejet 14. november, 2022
Odobren 2. december, 2022

2 PAMETNA QR-KODA

QR-koda je nadgradnja klasičnih črtnih kod, ki lahko vsebujejo le relativno majhno število informacij (približno 20 alfa numeričnih znakov). QR-koda z dodatkom nove dimenzije ustvari dvodimenzionalen matrični zapis, ki ima veliko večjo gostoto informacij na dani površini. Količina vsebovanih podatkov je odvisna od različnih standardov in parametrov. V osnovi je QR-koda razporeditev črnih točk na beli podlagi v kvadratni obliki. Poznamo več različic QR-kod, ki se delijo glede na velikost in posledično maksimalno možno količino podatkov, ki jih koda lahko vsebuje: od najmanjše različice 1, ki ima 21×21 slikovnih točk, pa vse do različice 40, ki lahko

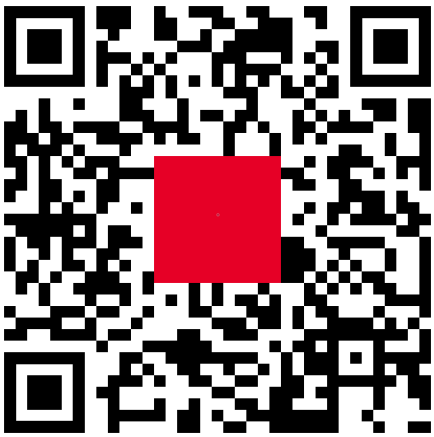
vsebuje največ informacij, v njej pa so podatki kodirani v matriki velikosti 177 × 177 slikovnih točk.

Slika 1: Različice QR-kode: na levi različica 1 (21 × 21), v sredini različica 2 (25 × 25) in desno različica 40 (177 × 177)

Odlična lastnost QR-kode je tudi odpravljanje napak. Za zagotovitev pravilnosti prebranih informacij obstajajo določeni razredi toleranc in korekcijskih faktorjev. S pomočjo **Reed-Solomonovega** algoritma se pri ustvarjanju QR-kode v njo dodajo kontrolna gesla, ki omogočajo branje informacije tudi iz delno zakrite ali poškodovane QR-kode. Ta pri najvišji stopnji H znaša do 30 % neberljive površine QR-kode. Ta "poškodovani"del pa mora biti zunaj obveznih polj, ki so namenjena informaciji o poziciji, poravnavi, ipd., ki bralniku QR-kode omogočajo, da jo prepozna. Na to lastnost lahko gledamo tudi drugače in sicer tako, da lahko v QR-kodi lahko mesta, ki niso nujno potrebna za razpoznavo osnovnih informacij, uporabimo za dodatno funkcijo. Taka "pametna"QR-koda poleg osnovnega zapisa podatkov vključuje še npr. barvno kodo [1], [2], [3].

3 Barvna koda

V del QR-kode, izbrano je bilo področje na sredini, kjer zagotovo ni obveznih sestavnih delov QR-kode, se je dodalo funkcionalizirano barvno področje, kjer se je informacija o temperaturnem območju gibanja izdelka shranila v odtenek določene barve. Primer take kode je prikazan na sliki 2.



Slika 2: QR-koda z osrednjim funkcionaliziranim področjem

Barva v kvadratu je izbrana glede na najboljšo sporočilno vrednost za človeka, glede na temperaturno območje ali alarmno sporočilo. Tako so modri odtenki največkrat izbrani za hladnejše barve, zeleni za področje okoli sobne temperature ter rdeči za višje temperature ter alarme. Največji izziv je nameščanje funkcionaliziranih barvnih področij na izdelke, ker morajo biti ti nameščeni v (za izdelek) ustreznem temperaturnem področju. To pomeni za zmrznjene izdelke v hladilnicah oziroma mora biti funkcionalizacija tam ustrezno aktivirana.

Za izdelavo pigmentov se izkorišča posebna lastnost določenih materialov in sicer **termokromizem**. To je le eden izmed številnih tipov kromizma. Predstavlja proces spreminjanja odtenka barve pigmenta zaradi izpostavljenosti določeni temperaturi. Ta sprememba je lahko povratna ali nepovratna. Pri povratnem termokromizmu se izpostavljenemu delu ob povratku na izhodiščno temperaturo povrne tudi osnovna barva, pri nepovratnem pa se sprememba barve ohrani. Še več, iz odtenka barve lahko pri določenih pigmentih natančno določimo doseženo temperaturo. To lastnost v praksi izkoriščamo za aplikacije, kot na primer na skodelicah kot pokazatelja temperature napitka, pri baterijah kot indikatorja napolnjenosti baterije in podobno. V uporabi sta dve glavni skupini materialov, ki se izrabljajo za termokromni učinek, prva so tekoči kristali in druga so levko barvila [4], [5].

Z razvojem temperaturno odvisnih pigmentov se v Sloveniji ukvarja podjetje **Mycol d.o.o.** [6], ki po želji uporabnika razvije, testira in predlaga najugodnejšo rešitev za specifično transportno pot izdelka. Tam se je tudi porodila ideja za razvoj pametne QR-kode in bralnika, ki bi omogočal zajem in nadalnjo uporabo podatkov ter po potrebi odziv operaterja.

4 Sistem za zaznavo in obdelavo pametne QR-kode

Glavni del sistema je razvojna platforma **ESP32-CAM** [7], [8], prikazana na sliki 3. Ta vsebuje dvojedrni procesor (frekvenca ure do 240 MHz), vgrajen vmesnik za kamero, WiFi in Bluetooth ter še mnogo drugih funkcionalnosti. Uporabljena kamera **OV2640** omogoča največjo ločljivost 2 megapiksla (1600 × 1200 pikslov) in podpira različne slikovne formate [9]. Za programiranje je bilo uporabljeno razvojno okolje **Arduino** [10].

Sistem, ki deluje kot ročni optični čitalnik, poleg prej omenjenega modula ESP32 vsebuje še druge pomembne dele:

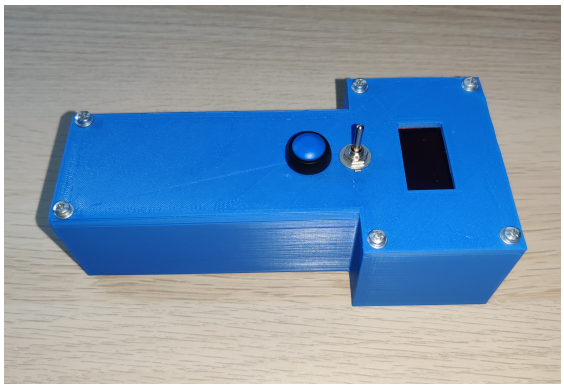
- **Zaslon** diagonale 1,3 palca z 128 × 64 slikovnimi pikami za prikaz rezultata zajema. Za upravljanje prikaza zaslon uporablja gonilnik SSD1306 ter komunikacijski protokol I2C v načinu "slave only", saj zaslon podatke za prikaz le sprejema, ne pa oddaja. Protokol I2C potrebuje 2 podatkovni liniji. To sta SCK (Serial clock), po kateri se prenaša

Slika 3: ESP32-CAM modul

ura, ter SDA (Serial Data), po kateri se serijsko prenašajo podatki. Za enostavno upravljanje zaslona sta uporabljeni dve vnaprej pripravljeni knjižnici. Prva je Adafruit_SSD1306 za upravljanje zaslonov na osnovi gonilnika SSD1306 [11], druga je Adafruit-GFX-Library za enostavno ustvarjanje grafičnih elementov na zaslonu [12].

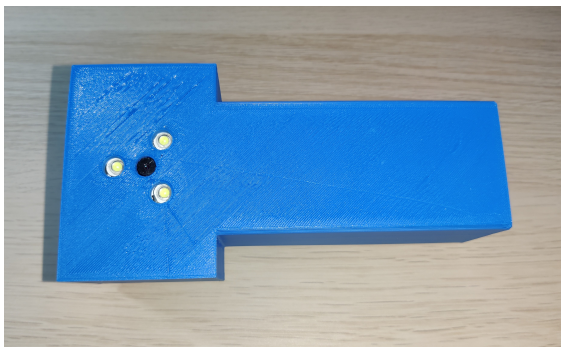
- **Osvetlitev** je realizirana s tremi visoko svetilnimi LED-diodami. Te za napajanje potrebujejo napetost 3,2-3,4 V ter prenesejo tok do 350 mA. Oddajajo svetlobo hladno bele barve (7000-9000 K) in sevajo 100-110 lumnov. Tri diode so vezane zaporedno za največjo skupno moč 3 W. Za napajanje svetlečih diod potrebujemo vir konstantnega toka, za kar je uporabljen tokovni regulator LD24AJTA na osnovi integriranega vezja **PT4115** [13], ter nekaj dodanh pasivnih elementov. Ta nam poleg vira konstantnega toka omogoča tudi upravljanje pulzno širinske modulacije za nastavljanje želene svetilnosti.
- **Napajanje** je zagotovljeno z dvema akumulatorskima baterijama **INR18650-29E**. Nazivna kapaciteta posamezne baterije je 2.850 mAh, nazivna napetost pa 3,7 V. Bateriji sta vezani vzporedno, s čimer se ohrani napetost, podvoji pa se skupna kapaciteta sistema, kar omogoča daljši čas delovanja naprave med posameznimi polnjenji. Za polnjenje baterij potrebujemo konstantno napetost 4,2 V ter kontroliran tok. Za ta namen je za polnjenje baterij dodano vezje na osnovi namenskega integriranega vezja **TC40-56A** [14]. Za enostavnejšo uporabo vezje vsebuje tudi priključek tipa USB-C, prek katerega lahko z ustreznim pretvornikom polnimo baterije. Poleg funkcije polnjenja pa to vezje vsebuje še nekatere funkcije za zaščito baterij, kot so izklop polnjenja ob napetosti baterije 4,2 V, omejitev največjega izhodnega toka ter zaščita proti prekomernim izpraznjenjem baterij. Ker ESP32 potrebuje 5 V stabilne napetosti, to zagotovimo iz baterijske napetosti 3,7-4,2 V s pomočjo vezja **MT3608** [15], ki deluje kot napetostni pretvornik navzgor. Podobno z istim vezjem storimo tudi za napajanje svetlečih diod, ki za delovanje potrebujejo 12 V napetosti.
- Preostali sestavni deli, kot so gladilni kondenzator 1000 µF na napajalnih linijah, stikalo za vklop in izklop napajanja, ki je vezano med baterije, ter napajalni priključek ESP32-CAM in gumb v kombinaciji s ”pull-down” uporom. Ob pritisku na ta gumb se na ustreznem priključku ESP32-CAM pojavi visok napetostni nivo (5 V), ki sproži branje QR-kode.

Vsi našteti moduli so bili testirani ločeno, nato pa povezani v sistem ter nameščeni v zato načrtano in s 3D-tiskalnikom izdelano plastično ohišje, predstavljeno na sliki 4.



Slika 4: Ohišje ročnega bralnika QR-kode - pogled zgoraj

Bralnik ima tik pod zaslonom stikalo za vklop. Ob vklopu se bralnik inicializira in se prek omrežja Wi-Fi poveže na strežnik ter stanje sistema sporoči na zaslonu. S tem je pripravljen na zajem slike QR-kode. Ob pritisku na tipko, tik pod stikalom za vklop, se sproži vklop osvetlitve, nato pa ESP32 prek kamere zajema sliko, neposredno pod bralnikom. V primeru pravilne detekcije QR-kode izklopi zajem in osvetlitev ter predstavi zajete podatke na zaslonu, hkrati pa podatke posreduje tudi na strežnik. Na sliki 5 je predstavljen spodnji pogled bralnika s svetlečimi diodami in kamero na sredini.

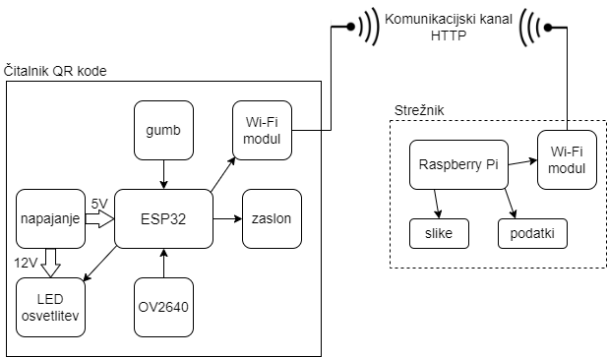


Slika 5: Ohišje ročnega bralnika QR-kode - pogled spodaj

5 DELOVANJE

Blokovna shema sistema, prikazana na sliki 6, predstavlja osnovne sestavne dele bralnika in komunikacijo s

strežnikom.

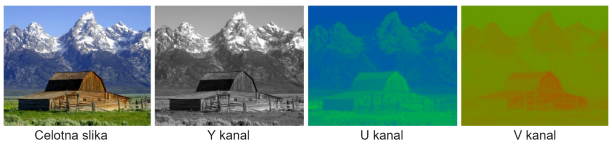


Slika 6: Blok shema sistema

Osnovna funkcija, ki jo mora bralnik opravljati, je torej razpoznava in dekodiranje vsebine QR-kode. Zato je uporabljena prosto dostopna programska knjižnica **ESP32QRCodeReader** [16], ki združuje tudi nekatere druge knjižnice, npr. za zajem slike, razpoznavanje vzorcev in matematične operacije. Ta knjižnica je služila za osnovo, treba jo je bilo le prilagoditi našim potrebam.

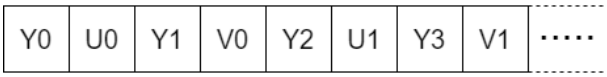
Prvi del programske kode in knjižnice služi za inicializacijo sistema. Najprej se ponastavi zaslon, nato nanj izpišemo pozdravno sporočilo. Sledijo nastavitev parametrov pulzno širinske modulacije za krmiljenje osvetlitve s svetlečimi diodami, deklaracija priključka za gumb in zagon povezave z omrežjem Wi-Fi. V kodi podamo uporabniško ime in geslo lokalnega omrežja ter se v zanki povezujemo na omrežje, dokler nam to ne uspe. Na koncu se ponastavi še kamera. V primeru napake pri opisanih korakih, operater to informacijo prejme prek zaslona, sistem pa je treba ponovno zagnati.

Ena izmed težav je, da se pri osnovnih nastavitvah na kameri izbere črno-bel zajem slike ločljivosti 320×240 slikovnih točk, saj to zahteva uporabljena knjižnica za razpoznavo QR-kode. Knjižnica zato potrebuje dodatno razširitev za barvni zajem, kar izbrana kamera omogoča. Izbran je bil **YUV** zapis, ki poleg barvnih kanalov vsebuje tudi ločen kanal Y za intenziteto posamezne slikovne točke, ki je pomembna za zaznavo temperaturne spremembe pigmenta. Obstoječa knjižnica je bila ustrezno nadgrajena za tak zajem. Ta format uporablja barvni prostor **Y-Cb-Cr** (ang.Y-luma, Chroma blue, Chroma red). Pri zajemu pa je bila narejena še poenostavitev, saj je človeško oko mnogo bolj občutljivo za spremembe v osvetljenosti kot za spremembe v barvi, zato barve lahko podvzorčimo. Dopolnjena oznaka takega formata je **YUV422**, kar pomeni, da sta barvna kanala U in V podvzorčena s faktorjem 2 - dve sosednji slikovni točki si delita barvni komponenti U in V, komponenta Y pa ostaja unikatna za vsako slikovno točko. To pomeni, da sta za posamezno slikovno točko potrebna le dva podatka. Zajem slike v izbranem formatu v programski kodi določimo pri ponastavitvi kamere z izbiro `cameraConfig.pixel_format = PIXFORMAT_YUV422`. Primer razdelitve osnovne slike na kanale YUV je predstavljen na sliki 7.



Slika 7: Primer barvne slike v formatu YUV ter posamezni kanali

Pri zajemu slike se torej ustvari tabela z dolžino enako $visina \times sirina \times 2$, kjer faktor 2 predstavlja dva podatka na slikovno točko. Primer podatkov je prikazan na sliki 8.



Slika 8: Zaporedje podatkov pri formatu YUV422.

Iz predstavljenih podatkov je treba izluščiti sivinsko sliko, ki jo kot osnovo uporablja funkcija za razpoznavo QR-kode, kar enostavno izvedemo tako, da uporabimo vsak drug podatek iz zgoraj prikazane tabele. To je v kodi izvedeno z zanko, ki pregleda vse elemente tabele in vsak drug element prepiše v novo tabelo. V programski kodi je ta pretvorba realizirana na naslednji način:

```
size_t new_len = fb1->len/2;
uint8_t *buf_grey = (uint8_t *) malloc(
    new_len);
camera_fb_t fb_grey_struct = {buf_grey,
    new_len, fb1->width, fb1->height,
    PIXFORMAT_GRAYSCALE, fb1->timestamp};
camera_fb_t *fb = &fb_grey_struct;
for(int i=0; i<new_len; i++){
    fb->buf[i] = fb1->buf[2*i];
}
```

`fb1` je struktura, ki vsebuje podatke za barvno sliko, `fb` pa je struktura, ki vsebuje podatke za sivinsko sliko.

Nato poskuša algoritem iz sivinske slike dekodirati vsebino QR-kode. Če mu to ne uspe, to prek zaslona javi uporabniku ter postopek ponovi, dokler je gumb za zajem pritisnjen oziroma dokler ni QR-koda uspešno dekodirana. Če je bilo dekodiranje uspešno, pa se podatki shranijo, algoritem nadaljuje prepoznavo barvnega dela. Ta se začne tako, da se najprej določi, kje na sliki dejansko je QR-koda, saj se pri dekodiranju zaznajo tudi koordinate položaja QR-kode. Te podatke ponovno uporabimo in s povprečenjem koordinat x in y določimo položaj središča QR-kode na sliki, kjer je dejansko funkcionalizirano področje. Te koordinate središča nato uporabimo in iz tabele elementov osnovne barvne slike izberemo podatke Y, U ter V, v področju 9×10 v okolici središča. S tem je zaznava barve nekoliko bolj zanesljiva. Izsek kode, ki poišče indekse podatkov slikovnih točk v zgornji vrstici izbranega pravokotnika, je naslednji:

```
uint32_t indeksi[180];
for(int j = 0; j<20; j++){
    indeksi[j] = center_index-640*4-4*2+j;
}
```

Spremenljivka `center_index` hrani lokacijo elementov enodimenzionalne tabele surovih podatkov slike, ki smo jo izračunali iz koordinat x in y središča QR-kode. Od te vrednosti najprej odštejemo 640×4, kar pomeni, da se premaknemo za 4 vrstice navzgor ($640 = 320$ slikovnih točk v vrstici $\times 2$ byta na slikovno točko), nato pa odštejemo še 4×2, da se premaknemo na zgornjo levo slikovno točko v pravokotniku. Podobno naredimo še za preostale vrstice. Po končanem postopku imamo v tabeli `indeksi` lokacijo vseh podatkov, ki jih bomo uporabili za razpoznavo barve.

S pomočjo te tabele najprej izluščimo posamezne komponente Y, U in V. Zopet pregledamo tabelo in iz nje izberemo naslove podatkov, ki nas zanimajo, nato pa te naslove uporabimo pri iskanju dejanskih podatkov YUV iz tabele v medpomnilniku. Pri tem upoštevamo vrstni red podatkov, kot je bilo prikazano na sliki 8.

```
for(int i = 0; i<90; i++){
    Y[i] = fb1->buf[(indeksi[2*i])];
}
for(int i = 0; i<45; i++){
    U[i] = fb1->buf[(indeksi[4*i+1])];
    V[i] = fb1->buf[(indeksi[4*i+3])];
}
```

Pred tem je seveda treba pripraviti tabele ustreznih velikosti, ki bodo te podatke hranile. Iz teh podatkov pa zdaj določimo povprečje, ki bo predstavljalo povprečno barvo na opazovanem področju. S tem zmanjšamo morebitne motnje na sliki ter ublažimo vpliv neenakomernosti osvetlitve. Rezultat je za lažje razumevanje operaterja oz. uporabnika pretvorjen v format **RGB**, informacija o intenziteti pa ostaja na voljo za nadaljnjo obdelavo. Rezultat zajema prikažemo na zaslonu bralnika, prav tako pa ga pošljemo tudi na strežnik, kamor se shrani tudi originalna zajeta barvna slika. Primer izpisa na zaslonu bralnika je prikazan na sliki 9.

Slika 9: Zaslon z informacijo o izdelku in zajeti barvi

Naloga strežnika je sprejemanje in hramba podatkov, prebranih iz QR-kod. Uporabljen strežnik je osnovan na platformi **Raspberry Pi** [17], potrebne skripte pa so napisane v jeziku PHP. Čitalnik prek povezave Wi-Fi z zahtevami HTTP POST pošilja podatke na strežnik. Raspberry Pi v vlogi strežnika mora biti povezan v isto lokalno omrežje, na katero je povezan tudi bralnik. Po vsakem uspešnem dekodiranju QR-kode bralnik najprej poskuša vzpostaviti komunikacijski kanal s strežnikom. Če je pri tem uspešen, se lahko začnejo nadaljnji koraki branja in shranjevanja. Poleg uporabniških podatkov moramo na strežnik poslati tudi nekatere druge informacije, da zna pravilno interpretirati prihajajoče podatke. Te informacije zapišemo v glavo, ki jo ločimo na dva dela - na glavo za tekst ter glavo za sliko. V tekstovni datoteki so zapisani podatki iz QR-kode. V delu glave, ki bo namenjen prenosu tega besedila, pa vnaprej pripravimo prostor za dodatno vrstico za zapis zajete barvne komponente RGB. Nato se izvede dejanski prenos podatkov na strežnik z zahtevami HTTP POST . Po besedilu se prenesejo še dejanski neobdelani podatki slike v formatu YUV, ki se shranijo v lastno datoteko za predvideno nadaljnjo obdelavo.

Na strani strežnika potrebujemo skripto, ki ustrezno obdela zahteve za pošiljanje podatkov. Najprej si pripravimo spremenljivke, ki vsebujejo ciljno mapo, imena datotek, datum in čas. Nato s serijo `if` stavkov preverimo, ali datoteka z enakim imenom že obstaja, ali je datoteka prevelika in ali je datoteka ustreznega tipa. Če karkoli ni pravilno, se to zazna in datoteka se ne shrani. Če pa so vsi podatki pravilni, pa nadaljujemo z zapisom besedilne in slikovne datoteke ločeno. Del kode za zapis tekstovne datoteke je naslednji:

```
file_put_contents($target_file_text, "
    --------".date('Y.m.d_H:i:s', $datum)."
    --------\n", FILE_APPEND);
$text = file_put_contents($target_file_text
    , file_get_contents($_FILES["
    information"]["tmp_name"]), FILE_APPEND
    );
file_put_contents($target_file_text, "\n",
    FILE_APPEND);
if($text == FALSE){
    echo "Sorry, there was an error
        updating your .txt file";
}
else{
    echo basename($_FILES["information"]["
        name"]). " has been updated.";
}
```

Zgornji odsek kode deluje v povezavi z delom glave, ki skrbi za tekstovni del. Ob prvem prenosu se na strežniku ustvari tekstovna datoteka `QR-information.txt`, v katero se zapišejo sprejeti podatki, to so podatki, prebrani iz QR-kode ter RGB-koda barve središčnega funkcionaliziranega področja. Za lažji pregled podatkov se dodata še datum in čas branja, v katerem je bila QR-koda zajeta. Ob vsakem nadaljnem uspešnem sprejemu podatkov na strežnik se novi podatki dodajo v isto besedilno datoteko. Nato se v novo datoteko, ki ji v ime prav tako dodamo datum in čas zajema, zapiše še surove barvne podatke slike QR-kode.

Da si lahko te slike dejansko ogledamo, strežnik opravi še pretvorbo surovih podatkov slike v `JPG` format. Ob vsakem ponovnem sprejemu podatkov strežnik najprej preišče mapo z datotekami, ki vsebujejo surove podatke o sliki, ter določi najnovejšo datoteko. To nato s pomočjo orodja ImageMagick [18] pretvorimo v običajno JPG-sliko, ki si jo po potrebi lahko ogledamo.

Več o pametni QR-kodi, uporabljeni programski kodi ter delovanju sistema kot celote, je opisano v diplomskem delu [19].

6 NADALNJI KORAKI

Predstavljen sistem za zajem je zgolj osnova za nadaljnji razvoj. Ena od glavnih nalog v prihodnosti je zagotoviti homogeno osvetlitev z visokim **CRI** faktorjem, ki nam zagotavlja osvetlitev, primerljivo z naravnim virom - soncem. Prav tako bi moral bralnik zagotavljati konstantno oddaljenost in enak kot za enako osvetlitev vseh kod. Veliko omejitev prinaša tudi ločljivost kamere, ki nam trenutno ne omogoča uporabe QR-kod višjih različic. Prav tako pa bi zmogljivejši senzor kamere omogočil natančnejši zajem barve in njenega odtenka. Trenutno za vse funkcije bralnika izkoriščamo le eno jedro sistema, na voljo pa sta dve jedri. Nalogo zajema in pošiljanja bi lahko ločili in s tem zagotovili prenos večje količine podatkov, brez prekinitev in napak, predvsem v primeru izpada brezžične povezave ter uporabe kamere z višjo ločljivostjo. Trenutno uporabljen strežnik, opravlja le osnovno nalogo hranjenja podatkov. Ta bi lahko podatke tudi aktivno obdeloval in se ob napakah odzval z alarmi. Predvsem pa bi morali podatke na strežniku zaščititi pred vdorom in krajo oziroma pred uničenjem. Ne nazadnje je možno nadgraditi tudi samo QR-kodo - z dodatnimi informacijami na preostalih mogočih mestih in jo izpopolniti, da bi zaznavala tudi pravilnost nanosa pigmenta ter se uporabljala za druge namene z drugače funkcionaliziranimi elementi (npr. magnetni zapis, osvetlitev, vlaga ipd.).

7 ZAKLJUČEK

V članku je predstavljena pametna QR-koda, ki ima funkcionaliziran osrednji del z barvnim poljem. Polje se v primeru uporabe na produktu prelepi z realnim pigmentnim nanosom, ki se odzove na spremembo temperature (termokromizem). Informacija o temperaturnem gibanju izdelka ni le v barvi, temveč predvsem v njenem odtenku, zato sta zajem in obdelava take kode zahtevnejša. Za branje je bil razvit ročni bralnik z LED-osvetlitvijo, ki omogoča zajem slike QR-kode in njeno obdelavo. Iz slike razbere podatke o izdelku ter ločeno informacijo o barvi in odtenku. Vse zajete informacije se prikažejo na zaslonu bralnika, zajeta osnovna slika ter podatki pa so hkrati posredovani tudi strežniku. Tako QR-koda kot bralnik sta že pripravljena za osnovno uporabo, treba ju je še nadgraditi za optimizacijo zajema in zaščito podatkov.

ZAHVALA

Zahvaljujem se podjetju MyCol za informacije v zvezi s termokromnimi pigmenti.

LITERATURA

[1] *QR Code Error Correction*, QRStuff.com, 2011.
[2] T.J. Soon, "QR code," *Synthesis journal*, pp. 59–78, 2008.
[3] S. Tiwari, "An Introduction to QR Code Technology," *2016 International Conference on Information Technology (ICIT)*, pp. 39–44, 2016.
[4] U. Bartol, *Izdelava pametne embalaže z vključitvijo indikatorja časa in temperature*, Univerza v Ljubljani, Naravoslovnotehniška fakulteta, 2020.
[5] O. Panák, B. Šumiga, B. Šumiga, P. Stražar, A. Sešek, M. Klanjšek Gunde, "Colour measurements of thermochromic indicators with various colouration temperatures," *Proceedings of the 2nd International Conference on Circular Packaging: Slovenj Gradec and online, 9th and 10th of September 2021*, pp. 261–268, 2021.
[6] *MyCol – Temperaturni nadzor vsakega živila po željah vsakega kupca*, http://www.mycol.si/sl/home/.
[7] *ESP32-CAM*, http://www.ai-thinker.com/pro_view-24.html.
[8] *ESP32-CAM Development Board*, https://media.digikey.com/pdf/Data
[9] "OV2640: Specs, Camera, Datasheet & Alternative (2022 Report)," Arducam, https://www.arducam.com/ov2640/.
[10] "Arduino Software," https://www.arduino.cc/en/software.
[11] "Adafruit SSD1306," https://github.com/adafruit/Adafruit_SSD1306.
[12] "Adafruit GFX Library," https://github.com/adafruit/Adafruit-GFX-Library.
[13] "30V, 1.2A Step-down High Brightness LED Driver with 5000:1 Dimming," https://www.electroschematics.com/wp-content/uploads/2014/07/PT4115E-datasheet.pdf.
[14] "TC4056A datasheet," https://datasheetspdf.com/pdf/1309136-/FUMANELECTRONICS/TC4056A/1.
[15] "MT3608 datasheet," https://datasheetspdf.com/pdf/909246/AEROSEMI/MT3608/1.
[16] "ESP32QRCodeReader," https://github.com/alvarowolfx/ESP32QRCodeReader.
[17] "Buy a Raspberry Pi 3 Model B," *Ltd, Raspberry Pi* hhttps://www.raspberrypi.com/products/raspberry-pi-3-model-b/.
[18] "PHP: ImageMagick - Manual," https://www.php.net/manual/en/book.imagick.php.
[19] R. Hrovat, *Razvoj in detekcija pametne QR kode*, Univerza v Ljubljani, Fakulteta za elektrotehniko, 2022.

Rok Hrovat je študent prvega letnika druge stopnje študija elektronike na Fakulteti za elektrotehniko, Univerze v Ljubljani. Njegovo področje raziskovanja zajema predvsem elektronska vezja, mikrokrmilnike ter strežnike Raspbery Pi.

Aleksander Sešek je docent na Fakulteti za elektrotehniko, Univerze v Ljubljani. Njegovo področje delovanja in raziskovanja so predvsem elektronski sistemi za zajem in obdelavo senzorskih signalov, analogna in mešana integrirana vezja ter visokofrekvenčni elementi ter sistemi.