

# Mikroupravljački sustav za otkrivanje požara

---

**Grgić, Matko**

**Undergraduate thesis / Završni rad**

**2023**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:200:332237>

*Rights / Prava:* [In copyright](#) / [Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-07-15**

*Repository / Repozitorij:*

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH  
TEHNOLOGIJA OSIJEK**

**MIKROUPRAVLJAČKI SUSTAV ZA OTKRIVANJE  
POŽARA**

**Završni rad**

**Matko Grgić**

**Osijek, 2023.**

# SADRŽAJ

<b>1. UVOD</b> .....	1
1.1. Zadatak rada .....	1
<b>2. PRIMJENJENE TEHNOLOGIJA I ALATI</b> .....	2
2.1. ESPDUINO-32 – mikroupravljačko sklopovlje .....	2
2.2. MLX90641 – Infracrveni senzor topline sa 16x12 matricom .....	2
2.3. Periferni uređaji .....	3
2.4. Arduino IDE .....	4
2.5. Visual Studio Code.....	5
<b>3. REALIZACIJA SUSTAVA</b> .....	6
3.1. Načini detekcije požara .....	6
3.2. Infracrveno zračenje .....	6
3.3. Razumijevanje i spajanje senzora.....	7
3.4. Izrada biblioteke .....	8
3.5. Programsko rješenje sustava.....	10
<b>4. EKSPERIMENT</b> .....	17
<b>5. ZAKLJUČAK</b> .....	19
<b>LITERATURA</b> .....	20
<b>SAŽETAK</b> .....	22
<b>ABSTRACT</b> .....	23
<b>ŽIVOTOPIS</b> .....	24
<b>PRILOZI</b> .....	25

# 1. UVOD

U suvremenom društvu, zaštita ljudi, imovine i okoliša od požara predstavlja izazov od velike važnosti. Upravljanje rizicima vezanim uz požare zahtijeva sve sofisticiranija tehnološka rješenja, a mikroupravljački sustavi postaju ključni elementi u ostvarivanju visoke razine sigurnosti. Mikroupravljački sustavi su integrirani sklopovi temeljeni na mikrokontrolerima ili mikroprocesorima, sposobni za precizno izvođenje niza zadataka u realnom vremenu. Njihova primjena u domenama kao što su otkrivanje požara omogućuje brzu, pouzdanu i inteligentnu reakciju na potencijalno opasne situacije.

Ovaj završni rad usredotočuje se na analizu, dizajn i implementaciju mikroupravljačkog sustava namijenjenog otkrivanju požara. Kroz sustavan pristup, istraživački rad istražuje različite aspekte vezane uz detekciju požara, uključujući različite tipove senzora za detekciju topline i plamena, metode obrade signala te algoritme za donošenje odluka.

U nastavku rada, detaljno je opisan proces projektiranja sustava, od odabira komponenata do implementacije softverskih algoritama. Također, proučeni su rezultati eksperimenata koji demonstriraju učinkovitost i pouzdanost razvijenog mikroupravljačkog sustava u kontekstu otkrivanja požara. Kroz ovu analizu, rad doprinosi sveobuhvatnijem razumijevanju uloge mikroupravljačkih sustava u poboljšanju sustava zaštite od požara te potiče daljnja istraživanja i inovacije u ovom važnom području.

U prvom poglavlju rada predstavljeni su osnovni ciljevi i motivacija za izradu završnog rada na zadanu temu. Drugo poglavlje daje detaljan prikaz svih korištenih alata i tehnologija nužnih za izradu rada dok treće poglavlje pruža pogled u detalje izrade fizičkog modela rada kao i programske podrške za isti. Četvrto poglavlje je opis održanog eksperimenta te rezultati i zaključci izvedeni iz eksperimenta.

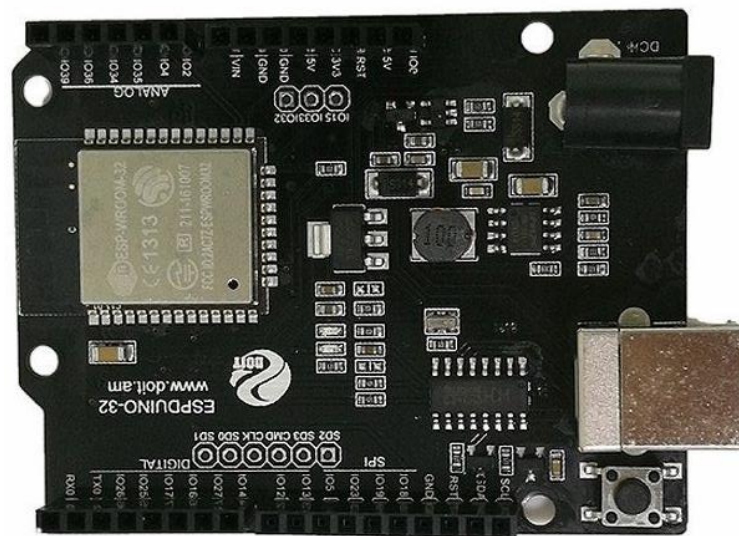
## 1.1. Zadatak rada

Zadatak rada je napraviti mikroupravljački sustav za otkrivanje požara. Glavni element rada je Melexis MLX90641 infracrveni senzor topline. Potrebno je razumjeti način njegovog djelovanja te ga povezati s mikroupravljačem ESPduino-32 tvrtke DOIT. Potrebno je osigurati i odgovarajuću programsku podršku te periferne uređaje. Na kraju sve zajedno povezati i omogućiti ispis na zaslon.

## 2. PRIMJENJENE TEHNOLOGIJA I ALATI

### 2.1. ESPDUINO-32 – mikroupravljačko sklopovlje

DOIT ESPduino-32 (Sl. 2.1.) je popularano mikroupravljačko sklopovlje koji se često koristi u različitim elektroničkim projektima. Njime upravlja Xtensa LX6 mikroprocesor, a uz njega na tiskanoj pločici se nalaze WiFi modul za bežični prijenos podataka i pristup internetu, Bluetooth 4.2 te modul za upravljanjem u stvarnom vremenu. Napaja se preko USB ulaza ili konektora istosmjerne struje na napon između 5 i 12V te strujom od 80mA [1].



Sl. 2.1. DOIT ESPduino-32 pločica s Xtensa LX6 mikrokontrolerom i perifernim uređajima [1]

### 2.2. MLX90641 – Infracrveni senzor topline sa 16x12 matricom

MLX90641 je industrijski standardiziran senzor topline s 16x12 matričnim prikazom, tj. 192 jedinstvena infracrvena senzora. Zapakiran je u TO-39 pakiranje s digitalnim sučeljem. Radi na 3.3V s jačinom struje od 12mA. Omogućava jednostavnu I2C komunikaciju brzine do 1MHz što ga čini pogodnim za jednostavne, ali i složenije elektroničke projekte. Neke od njegovih upotreba u stvarnom svijetu su: beskontaktno mjerenje temperature, mikrovalne pećnice, prepoznavanje pokreta, klima uređaji s automatskim paljenjem i sl. Senzor ima četiri pina – dva za I2C komunikaciju (SDA i SCL) te dva za spajanje samog uređaja u strujni krug [2].



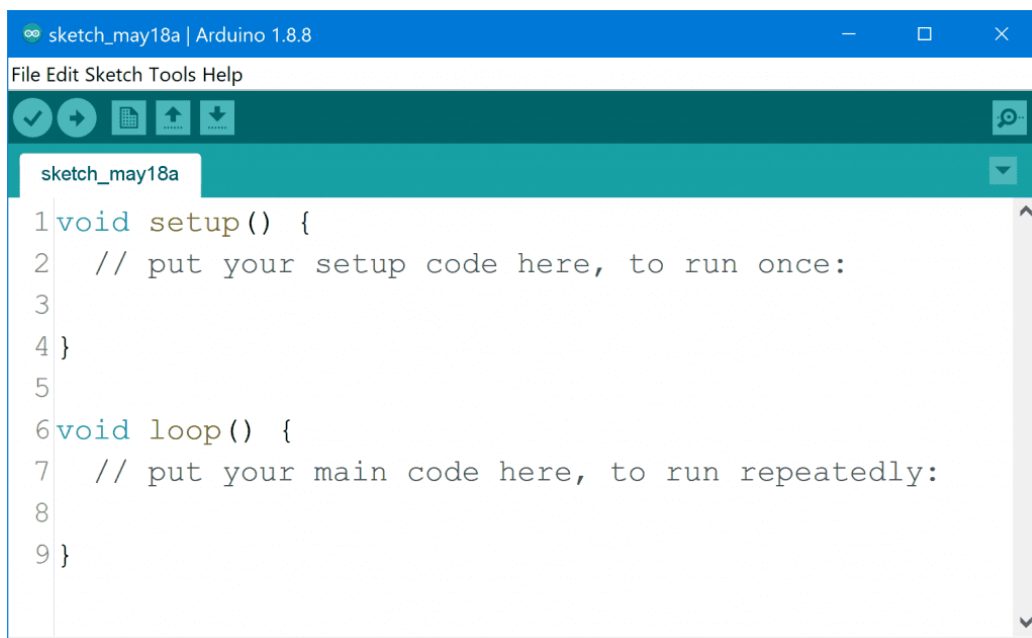
Sl. 2.2. MLX90641 senzor topline sa 16x12 matričnim prikazom [3]

### 2.3. Periferni uređaji

Niti jedan elektronički projekt nije moguć bez perifernih uređaja kao što su: otpornici, vodiči i diode. Za ovaj projekt korišteni su otpornici od 300 i 2200 $\Omega$  te LE dioda za označavanje alarmantnog stanja. Otpornici su nužni za održavanje sustava u stabilnom stanju – kada nema I2C komunikacije potrebno je pomoću pull-up otpornika dovesti logičku jedinicu na ulaze I2C komunikacije, a kad se događa prijenos podataka dolazi logička nula koja označava početak. Otpornici također služe za zaštitu ostalih perifernih uređaja, što je u ovom slučaju dioda, od pregrijavanja smanjujući struju koja prolazi kroz uređaj. Za realizaciju sustava korištena je i eksperimentalna pločica (*engl. breadboard*). Ova pločica idealna je za eksperimentalne sustave jer omogućuje spajanje komponenti na glavnu upravljačku jedinicu bez trajnog lemljenja komponenti. Sadrži velik broj „rupa“ (*engl. pins*) koji služe kao mjesto kontakta za vodiče. Redci pločice povezani su u seriju, pa se može gledati svaki redak kao zasebno mjesto kontakta. Ukratko, zbog svoje jednostavnosti eksperimentalne pločice često su korištenje za edukaciju o osnovama rada elektroničkih krugova, ali i za ozbiljnije eksperimentalne radove.

## 2.4. Arduino IDE

Arduino Integrated Development Environment (engl. *Arduino integrirano razvojno sučelje*) je program tvrtke Arduino poznat po svojim brojnim razvojnim pločicama. Ovo sučelje je programski dodatak tim, ali i mnogim drugim pločicama, koje omogućuje komunikaciju s hardverom, programiranje istog te njegovo testiranje. Sadrži uređivač teksta, prozor za izlaz, Serial monitor – korisni alat za provjeru rada koda preko fizičkog mikroupravljača i brojne druge mogućnosti. Budući da je sučelje kroz godine nadograđivano kako bi podržalo brojne uređaje, razvojni inženjeri su morali pisati svoje biblioteke kodova za svoje sklopovlje te je i gotove biblioteke moguće instalirati preko ugrađenog pretraživača. Dvije su glavne funkcije u kodu pisanog u Arduino IDE – `setup()` i `loop()`. `Setup()` (engl. *postavljanje*) je funkcija koja se izvodi samo jednom i poziva se na početku izvođenja programa. U njemu se govori koji izlazi se koristi i u koju svrhu, definiranje biblioteka, inicijalizacija varijabli i serijske komunikacije i sl. Nakon što su određeni početni uvjeti izvodi se `loop()` (engl. *ponavljanje*) funkcija. Kao što joj ime kaže, sav kod u njoj se ponavlja sve dok ne dođe do prekida programa. Na taj način moguće je čitati nove podatke koje senzor prikuplja, paliti i gasiti LED-icu i dr. Arduino programsko sučelje podržano je na svim većim platformama uključujući Windows, MacOS i Linux. [4]

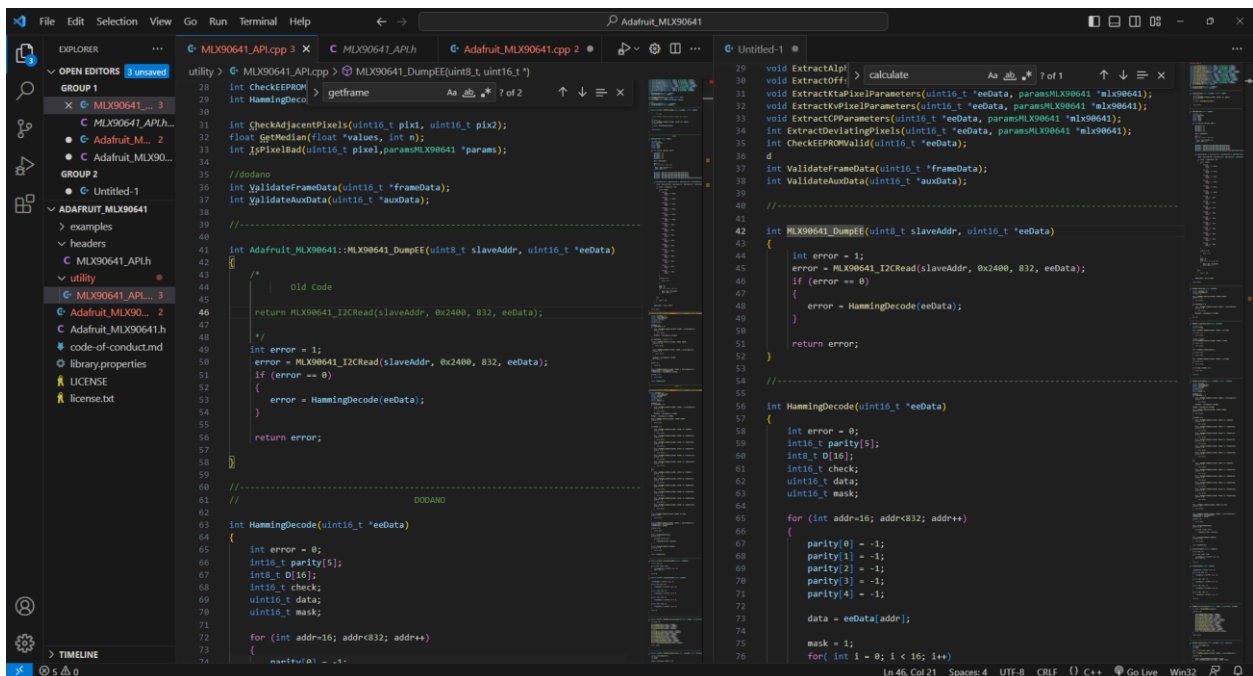


```
sketch_may18a | Arduino 1.8.8
File Edit Sketch Tools Help
sketch_may18a
1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```

Sl. 2.4. Arduino integrirano razvojno sučelje [5]

## 2.5. Visual Studio Code

Visual Studio Code (VS Code) je široko korišteni alat tvrtke Microsoft koji služi za uređivanje teksta, odnosno koda. Uz svoju potporu za brojne programske i opisne jezike, jednostavan je i intuitivan, ali i vrlo brz što ga čini povoljnim za širok spektar korisnika. Sadrži i inteligentna svojstva koja omogućuju samostalno završavanje započetih kodova, ali i predlaže sljedeće korake u pisanju programa. Ono što ga odvaja od ostalih alata slične prirode je njegova tzv. trgovina dodatcima. Slično kao i Arduino, postoje mnoga proširenja uključena u program od strane zajednice programera, kao i samog Microsofta, koja olakšavaju pisanje, uređivanje i razumijevanje koda. Program je besplatan, redovno nadograđivan i pouzdan što ga izdvaja kao jednog od najboljih alata u svojoj kategoriji. [6]



```
28 int CheckEEPROM
29 int HammingDeco
30
31 int (checkAdjacentPixels(uint16_t pix1, uint16_t pix2);
32 float getMedian(float *values, int n);
33 int (getPixelIAD(uint16_t pixel, paramsMLX90641 *params);
34
35 //dodano
36 int (validateFrameData(uint16_t *frameData);
37 int (validateAuxData(uint16_t *auxData);
38
39 //-----
40
41 int Adafruit_MLX90641::MLX90641_DumpEE(uint8_t slaveAddr, uint16_t *eeData)
42 {
43     // Old Code
44     return MLX90641_I2CRead(slaveAddr, 0x2400, 832, eeData);
45     //
46     int error = 1;
47     error = MLX90641_I2CRead(slaveAddr, 0x2400, 832, eeData);
48     if (error == 0)
49     {
50         error = HammingDecode(eeData);
51     }
52     return error;
53 }
54
55 //-----
56 // DODANO
57
58 int HammingDecode(uint16_t *eeData)
59 {
60     int error = 0;
61     int16_t parity[5];
62     int8_t D[16];
63     uint16_t check;
64     uint16_t data;
65     uint16_t mask;
66     for (int addr=16; addr<832; addr++)
67     {
68         parity[0] = -1;
69         parity[1] = -1;
70         parity[2] = -1;
71         parity[3] = -1;
72         parity[4] = -1;
73         data = eeData[addr];
74         mask = 1;
75         for (int i = 0; i < 16; i++)
```

Sl. 2.5. Visual Studio Code uređivač koda



### 3. REALIZACIJA SUSTAVA

#### 3.1. Načini detekcije požara

Postoji puno načina za detekciju požara, ali četiri su najčešća: ionizacijsko-fotoelektrični, fotoelektrični, ionizacijski i toplinski. Dok su toplinski senzori aktivirani povećanom temperaturom, ostali pale alarmantno stanje detekcijom dima [7]. Za ovaj rad korišten je toplinski senzor koji pomoću infracrvenog zračenja daje matrični prikaz temperatura svoje okoline.

#### 3.2. Infracrveno zračenje

Radi lakšeg razumijevanja rada senzora potrebno je upoznati se s osnovama infracrvenog zračenja. Infracrveno zračenje vrsta je elektromagnetskih valova valne duljine između 0.75  $\mu\text{m}$  i 1 mm. Zasluge za otkriće ove vrste zračenja upisuju se Fredericku Williamu Herschelu, engleskom astronomu, graditelju teleskopa i glazbeniku njemačkog podrijetla [8].

Promatrajući disperziju sunčeve svjetlosti na optičkoj prizmi primijetio je kako se više temperature nalaze na području crvenog dijela vidljivog spektra. Ta se vrsta zračenja ne može vidjeti, kao ni veći dio spektra elektromagnetskih valova, ali može ga se osjetiti na koži kao osjećaj topline. Infracrveno zračenje nastaje kao posljedica sudaranja čestica prilikom gibanja u prostoru, pri vibracijama kristalne rešetke čvrstih tijela te rotaciji kemijskih veza atomskih skupina u molekulama organskih tvari i plinova na svim temperaturama višim od apsolutne nule na kojoj svako gibanje staje [9]. Infracrveno zračenje, stoga, može se pojednostaviti i kao prikaz temperature nekog tijela ili plina.



Sl. 3.1. Toplinska (termalna) fotografija dviju osoba [10]

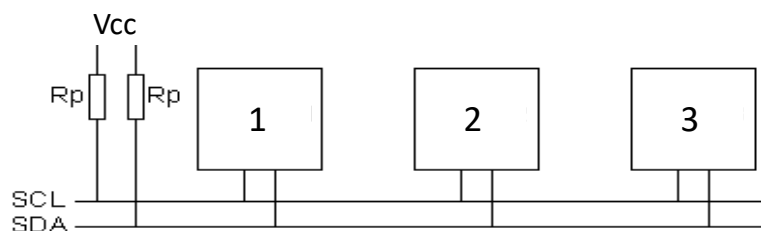
Na sl. 3.1. prikazana je toplinska fotografija. Koristeći srednje i duže vrijednosti valne duljine u infracrvenom spektru senzor „osjeti“ razliku topline. Tako je osoba, koja daje više topline od npr.

betonskog zida, prikazana svjetlijim (toplijim) bojama. Ova tehnologija često se koristi u sigurnosnim sustavima koji osiguravaju područje bez svjetla, primjerice vanjske prostore noću te područje s preprekama (šuma, polja s visokim raslinjem).

### 3.3. Razumijevanje i spajanje senzora

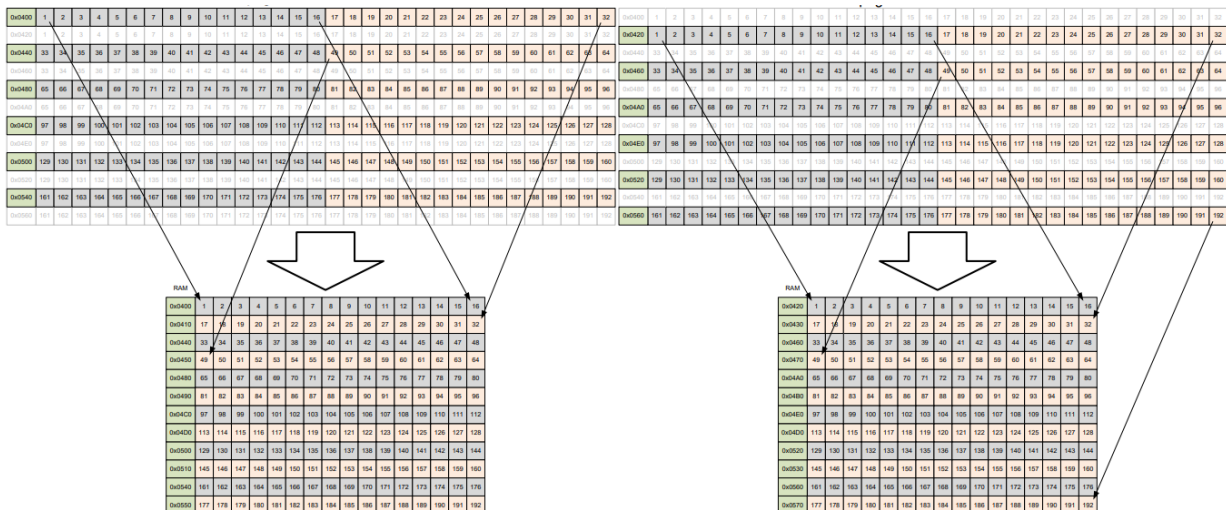
Prvi korak u izradi ovog rada je dobro se upoznati s glavnom komponentom. MLX90641 infracrveni senzor sa 16x12 matričnim prikazom okoline odabran je zbog svog jednostavnog izgleda i mogućnosti lakog rukovanja. Nalazi se u industrijski standardiziranom TO39 pakiranju – kućište napravljeno u obliku metalne kape, hermetički zatvoreno kako bi se mikročip zaštitio od okolnih utjecaja kao što su vlaga i zagađenje [11]. MLX90641 sadrži ambijentalni temperaturni senzor koji očitava vrijednosti temperature okoline spremajući vrijednosti u integriranu RAM memoriju, a kojoj pristupamo preko I2C protokola. Senzor ima 4 pina (nožice) preko kojih ga se napaja i komunicira s mikroupravljačkim sučeljem. Dvije nožice služe za napajanje dok druge dvije omogućavaju komunikaciju preko I2C protokola [2].

I2C protokol omogućuje mikrokontroleru komunikaciju s više različitih uređaja preko samo dvije žice – SDA i SCL. SCL (Serial Clock Line) stvara signal takta. Pri velikim brzinama koje se ostvaruju u svijetu elektronike svaka milisekunda je bitna. SCL sinkronizira, tj. ujednačuje, prijenos svih podataka u I2C kanalu kako bi signal istovremeno dolazio do svakog uređaja spojenog u taj kanal. SDA (Serial Data Line) služi za prijenos podatka od mikrokontrolera do uređaja i obratno – moguća je komunikacija u oba smjera, ali samo u jednom istovremeno. Uređaj koji šalje zahtjev nazivamo master, a onaj koji mu je podređen slave. Master šalje adresu uređaja s kojim želi komunicirati, uređaj koji ima tu adresu šalje/prima podatke dok ostali uređaji ostaju u stanju u kojem su prethodno bili. Za pravilnu I2C komunikaciju potrebno je na SCL i SDA linije postaviti pull-up otpornike. Oni spriječavaju da se u kanalu pojavi neodređeno stanje ukoliko se komunikacija trenutno ne odvija tako što spajaju linije s logičkom jedinicom („visokom“ naponskom razinom). Potrebno je razumjeti da prilikom spajanja elektroničkih potrebno je točno odrediti koje stanje se nalazi na izlazu (logička jedinica ili nula) kako bi se izbjegle pogreške [12].



Sl. 3.2. I2C komunikacija preko SCL i SDA linije i pripadajućim pull-up otpornicima [13]

MLX90641 senzoru potrebno je dovesti napon iznosa 3-3.6V i struju 10-14mA. I2C komunikacija radi na signalu taktu do 1MHz. Tvornička I2C adresa je 0x33 i potrebno ju je zapamtiti kako bi mogli programirati senzor. Sadrži 192 infracrvena senzora, tj. 192 vrijednosti temperature koje se prikazuju u matrici 16x12. Svaki piksel identificiran je svojim redom i stupcem kao Pix(i,j). Senzor sadrži i dvije stranice (*engl. subpage*) s kojih naizmjenično čita podatke.



Sl. 3.3. Stranice podataka unutar MLX90641 senzora [2]

Na Sl.3.3. prikazano je naizmjenično osvježavanje podataka po redcima – budući da se radi s 32 bitnim adresama potrebno je 32x6 prikaz transformirati u 16x12 koji je puno lakši za razumjeti i vizualizirati.

### 3.4. Izrada biblioteke

Biblioteka predstavlja kolekciju unaprijed napisanih kodova. Njihova svrha je ponuditi korisniku široku lepezu funkcija i mogućnosti koje može koristiti bez da ih osobno piše. Često se radi o kompliciranim funkcijama koje komuniciraju s operacijskim sustavom ili harverom, a razumijevanje istih nije nužno za daljnji rad programa. Kao što je ranije rečeno, Arduino IDE podržava biblioteke za velik broj senzora koji imaju sebi specifičan način rada. Često korišteni senzori i mikroupravljači imaju bolju podršku i unaprijed izrađene biblioteke od strane programera koji se time bave [14]. MLX90641 jedan je od onih koji nemaju svoju vlastitu biblioteku te je potrebno izraditi je iznova.

Tvrtka Melexis bavi se proizvodnjom elektroničkih komponenti među kojima je i MLX90641, ali i brojni drugi. Jedan od njihovih senzora, MLX90640, vrlo je sličan MLX90641

senzoru koji koristimo u projektu, a za njega postoje već unaprijed definirane biblioteke u Arduino IDE. U ovom slučaju, one su poslužile kao baza za izradu nove biblioteke.

Prvo je potrebno pomno proučiti razlike između dva senzora. Imaju gotovo istu namjenu, oba su matični infracrveni senzori topline, ali različitih rezolucija – MLX90640 sadrži matricu dimenzija 32x24. Zbog toga je prvo potrebno ući u svaku datoteku biblioteke i promijeniti dimenzije, uvjete petlji, adrese i sl. Za to je korišten VS Code uređivač koda koji može otvoriti cijele mape i omogućiti vrlo pregledan prikaz datoteka u njima. Zatim, potrebno je dodati Hammingov dekoder koji je nepotreban za rad MLX90640 senzora, ali iznimno potreban za rad MLX90641 senzora. Hammingov kod koristi se kao alat za pronalaženje i ispravljanje grešaka prilikom prijenosa podataka. Zahtjeva dodavanje paritetnih bitova čija se vrijednost određuje prema već postojećim bitovima informacije koja se prenosi. U ovom slučaju koristimo Hamming(15,11) kodiranje gdje od 11 podatkovnih bita dobivamo 4 paritetna bita. Peti paritetni bit dobiva se kao kombinacija svih podatkovnih bitova i svih paritetnih bitova dobivenim Hamming kodiranjem.

$$P0 = D0 + D1 + D3 + D4 + D6 + D8 + D10$$

$$P1 = D0 + D2 + D3 + D5 + D6 + D9 + D10$$

$$P2 = D1 + D2 + D3 + D7 + D8 + D9 + D10$$

$$P3 = D4 + D5 + D6 + D7 + D8 + D9 + D10$$

$$P4 = D0 + D1 + D2 + D3 + D4 + D5 + D6 + D7 + D8 + D9 + D10 + P0 + P1 + P2 + P3$$

Formule 3.1. Računanje vrijednosti paritetnih bitova [2]

B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
P4	P3	P2	P1	P0	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Sl. 3.4. Raspored podatkovnih i paritetnih bitova dobivenih Hamming kodiranjem te način kodiranja [2]

Izostavljanjem Hammingovog kodiranja, odnosno dekodiranja podataka, dobivaju se nasumične pozitivne i negativne vrijednosti, a često dolazi i do podataka koji nisu brojevi (NaN *engl. Not a Number*).

Nadalje, MLX90640 sadrži dva moda rada. Učitavanje podataka moguće je na jednak način kao i kod MLX90641 – čitanje svih vrijednosti iz „parnih“ heksadecimalnih adresa u jednom signalu takta, te čitanje svih vrijednosti iz „neparnih“ adresa u drugom taktu. To je

takozvani TV mod ili interleave mod rada. MLX90640 može raditi i na način da u svakoj stranici osvježava istovremeno sve adrese, ali i ne sve podatke u njima u obliku šahovnice (šah mod – *engl. Chess mode*).

0x0400	1	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31
0x0420	34	67	100	133	166	199	232	265	298	331	364	397	430	463	496	529
0x0440	66	131	196	261	326	391	456	521	586	651	716	781	846	911	976	1041
0x0460	98	196	294	392	490	588	686	784	882	980	1078	1176	1274	1372	1470	1568
0x0480	129	257	385	513	641	769	897	1025	1153	1281	1409	1537	1665	1793	1921	2049
0x04A0	162	324	486	648	810	972	1134	1296	1458	1620	1782	1944	2106	2268	2430	2592
0x04C0	193	387	580	774	968	1162	1356	1550	1744	1938	2132	2326	2520	2714	2908	3102
0x04E0	226	451	676	901	1126	1351	1576	1801	2026	2251	2476	2701	2926	3151	3376	3601
0x0500	257	514	771	1028	1285	1542	1800	2057	2314	2571	2828	3085	3342	3600	3857	4114
0x0520	290	580	870	1160	1450	1740	2030	2320	2610	2900	3190	3480	3770	4060	4350	4640
0x0540	321	642	963	1284	1605	1926	2247	2568	2889	3210	3531	3852	4173	4494	4815	5136
0x0560	354	708	1062	1416	1770	2124	2478	2832	3186	3540	3894	4248	4602	4956	5310	5664
0x0580	385	770	1160	1550	1940	2330	2720	3110	3500	3890	4280	4670	5060	5450	5840	6230
0x05A0	418	836	1254	1672	2090	2508	2926	3344	3762	4180	4598	5016	5434	5852	6270	6688
0x05C0	449	898	1347	1830	2313	2796	3279	3762	4245	4728	5211	5694	6177	6660	7143	7626
0x05E0	482	964	1454	1968	2482	2996	3510	4024	4538	5052	5566	6080	6594	7108	7622	8136
0x0600	513	1026	1542	2070	2604	3138	3672	4206	4740	5274	5808	6342	6876	7410	7944	8478
0x0620	546	1092	1644	2196	2760	3324	3888	4452	5016	5580	6144	6708	7272	7836	8400	8964
0x0640	577	1154	1734	2310	2904	3498	4092	4686	5280	5874	6468	7062	7656	8250	8844	9438
0x0660	610	1220	1830	2436	3060	3690	4320	4950	5580	6210	6840	7470	8100	8730	9360	9990
0x0680	641	1282	1932	2568	3216	3864	4512	5160	5808	6456	7104	7752	8400	9048	9696	10344
0x06A0	674	1348	2040	2712	3384	4068	4752	5436	6120	6804	7488	8172	8856	9540	10224	10908
0x06C0	705	1410	2154	2874	3570	4278	4986	5694	6402	7110	7818	8526	9234	9942	10650	11358
0x06E0	738	1476	2274	3048	3780	4536	5292	6048	6804	7560	8316	9072	9828	10584	11340	12096
0x0700	771	1542	2400	3228	4008	4800	5592	6384	7176	7968	8760	9552	10344	11136	11928	12720
0x0720	804	1608	2532	3420	4224	5040	5856	6672	7488	8304	9120	9936	10752	11568	12384	13200
0x0740	837	1674	2664	3540	4380	5220	6036	6852	7668	8484	9300	10116	10932	11748	12564	13380
0x0760	870	1740	2796	3672	4536	5436	6252	7068	7884	8700	9516	10332	11148	11964	12780	13596
0x0780	903	1806	2928	3810	4680	5604	6420	7236	8052	8868	9684	10500	11316	12132	12948	13764
0x07A0	936	1872	3060	3960	4848	5804	6624	7440	8256	9072	9888	10704	11520	12336	13152	13968
0x07C0	969	1938	3204	4116	5016	5984	6804	7620	8436	9252	10068	10884	11700	12516	13332	14148
0x07E0	1002	2004	3354	4272	5172	6168	7004	7836	8668	9492	10316	11140	11964	12788	13612	14436
0x0800	1035	2070	3504	4434	5340	6360	7200	8040	8880	9720	10560	11400	12240	13080	13920	14760
0x0820	1068	2136	3654	4596	5520	6564	7416	8264	9112	9960	10808	11656	12504	13352	14200	15048
0x0840	1101	2202	3804	4758	5688	6756	7620	8472	9324	10176	11028	11880	12732	13584	14436	15288
0x0860	1134	2268	3954	4920	5880	6972	7848	8700	9552	10404	11256	12108	12960	13812	14664	15516
0x0880	1167	2334	4104	5094	6096	7200	8076	8928	9780	10632	11484	12336	13188	14040	14892	15744
0x08A0	1200	2400	4260	5268	6300	7428	8304	9156	10008	10860	11712	12564	13416	14268	15120	15972
0x08C0	1233	2466	4416	5442	6480	7632	8508	9348	10192	11036	11880	12724	13568	14412	15256	16100
0x08E0	1266	2532	4572	5616	6660	7836	8712	9540	10376	11212	12048	12884	13720	14556	15392	16228
0x0900	1299	2598	4728	5796	6840	8040	8916	9732	10560	11388	12216	13044	13872	14700	15528	16356
0x0920	1332	2664	4884	5976	7032	8256	9120	9936	10764	11592	12420	13248	14076	14904	15732	16560
0x0940	1365	2730	5040	6156	7272	8504	9360	10176	11004	11832	12660	13488	14316	15144	15972	16800
0x0960	1398	2796	5196	6336	7416	8664	9516	10320	11148	11976	12804	13632	14460	15288	16116	16944
0x0980	1431	2862	5304	6516	7560	8856	9708	10512	11336	12160	12984	13808	14632	15456	16280	17104
0x09A0	1464	2928	5460	6696	7716	9000	9888	10692	11520	12348	13176	14004	14832	15660	16488	17316
0x09C0	1497	2994	5616	6876	7872	9192	10080	10884	11712	12540	13368	14196	15024	15852	16680	17508
0x09E0	1530	3060	5772	7056	8040	9396	10296	11092	11916	12744	13572	14400	15228	16056	16884	17712
0x0A00	1563	3126	5928	7236	8204	9600	10504	11292	12120	12948	13776	14604	15432	16260	17088	17916
0x0A20	1596	3192	6084	7416	8364	9900	10716	11496	12324	13152	13980	14808	15636	16464	17292	18120
0x0A40	1629	3258	6240	7596	8532	10200	10932	11700	12528	13356	14184	15012	15840	16668	17496	18324
0x0A60	1662	3324	6396	7776	8700	10488	11172	11924	12752	13580	14408	15236	16064	16892	17720	18548
0x0A80	1695	3390	6552	7956	8872	10644	11384	12112	12940	13768	14596	15424	16252	17080	17908	18736
0x0AA0	1728	3456	6708	8136	9048	10800	11592	12300	13128	13956	14784	15612	16440	17268	18096	18924
0x0AC0	1761	3522	6864	8316	9216	10968	11784	12472	13300	14128	14956	15784	16612	17440	18268	19096
0x0AE0	1794	3588	7020	8496	9396	11124	12000	12684	13512	14340	15168	15996	16824	17652	18480	19308
0x0B00	1827	3654	7176	8676	9576	11280	12156	12840	13668	14496	15324	16152	16980	17808	18636	19464
0x0B20	1860	3720	7332	8856	9756	11436	12312	13004	13832	14660	15488	16316	17144	17972	18800	19628
0x0B40	1893	3786	7488	9036	9936	11592	12480	13168	14000	14832	15664	16496	17328	18160	18992	19824
0x0B60	1926	3852	7644	9216	10116	11748	12648	13336	14172	15004	15836	16668	17500	18332	19164	19996
0x0B80	1959	3918	7796	9396	10296	11904	12828	13516	14352	15184	16016	16848	17680	18512	19344	20172
0x0BA0	1992	3984	7952	9576	10476	12060	13000	13688	14524	15356	16188	17020	17852	18684	19516	20340
0x0BC0	2025	4050	8100	9756	10656	12216	13184	13872	14712	15544	16376	17208	18040	18872	19704	20532
0x0BE0	2058	4116	8256	9936	10836	12372	13272	13960	14800	15632	16464	17296	18128	18960	19792	20616
0x0C00	2091	4182	8400	10116	11016	12528	13368	14056	14892	15724	16556	17388	18220	19052	19884	20712
0x0C20	2124	4248	8556	10296	11196	12684	13560	14240	15076	15908	16740	17572	18404	19236	20068	20896
0x0C40	2157	4314	8700	10476	11376	12840	13752	14432	15268	16100	16932	17764	18596	19428	20260	21088
0x0C60	2190	4380	8856	10656	11556	13000	13944	14624	15460	16292	17124	17956	18788	19620	20452	21276
0x0C80	2223	4446	9000	10836	11736	13156	14136	14816	15652	16484	17316	18148	18980	19812	20644	21468
0x0CA0	2256	4512	9156	11016	11916	13312	14336	15000	15836	16668	17500	18332	19164	19996	20828	21648
0x0CC0	2289	4578	9300	11196	12096	13468	14536	15216	16052	16884	17716	18548	19380	20212	21044	21864
0x0CE0	2322	4644	9456	11376	12276	13624	14640	15320	16156	16988	17820	18652	19484	20316	21144	21960
0x0D00	2355	4710	9600	11556	12456	13780	14744	15424	16260	17092	17924	18756	19588	20420	21240	22056
0x0D20	2388	4776	9756	11736	12636	13936	14944	15624	16460	17292	18124	18956	19788	20620	21440	22248
0x0D40	2421	4842	9900	11916	12816											

```
#include <Adafruit_MLX90641.h>
```

### Programski kod 3.1. Uključivanje biblioteke u program

Nadalje, potrebno je definirati senzor kao varijablu. U biblioteci je definirana klasu **Adafruit\_MLX90641** i u njoj se nalaze sve funkcije koje se koriste s istom klasom. Korišten je operator izravnog odabira člana ili popularnije točkasti (engl. *dot*) operator kako bi se preko varijable pristupalo funkcijama. Za spremanje podataka prikupljenih sa senzora potrebno je polje decimalnih vrijednosti od 192 mjesta. Kasnije se iz tog polja slaže matrični prikaz 16x12 koji je mnogo jednostavniji za čitanje i razumijevanje od jednodimenzionalnog polja.

```
Adafruit_MLX90641 mlx;  
float frame[12*16]; // buffer for full frame of temperatures
```

### Programski kod 3.2. Deklariranje klase varijablom i polja za spremanje temperatura

MLX90641 pomoću svoja 192 infracrvena senzora omogućava određivanje temperature iz okoline, ali češća je praksa prikazivanje podataka bojama ili simbolima, nego brojčano. U ovom slučaju programski je omogućen prikaz pomoću ASCII simbola. Svaka vrijednost temperature koja pripada određenom intervalu zamijenjena je i prikazana odgovarajućim simbolom. Vrlo visoke temperature koje daje plamen postavljaju sustav u alarmantno stanje koje naizmjenično pali i gasi LE diodu označavajući uzbunu. Kako bi mikroupravljač mogao komunicirati s LE diodom potrebno je unaprijed definirati pin preko kojeg se komunicira. Također, korišten je i brojač visokih temperatura kao osigurač. Moguće je zbog nepravilnosti ili greške da vrijednost skoči (engl. *spike value*) i prebaci sustav u alarmantno stanje. Brojač broji koliko se visokih temperatura nalazi u polju te, ako taj broj prelazi unaprijed određenu graničnu vrijednost, postavlja sustav u alarmantno stanje.

```
// uncomment *one* of the below  
//#define PRINT_TEMPERATURES  
#define PRINT_ASCIIART  
const int pinToControl = 26;  
int count = 0;
```

### Programski kod 3.3. Definicija načina rada, definiranje komunikacijskog pina za LE diodu te deklaracija i inicijalizacija brojača

```

xx. .x+. .xx--#%**
%x. .x+--xx-+#X**
xx. -+x--x+-*##**
%x-*xx--+-xx-*
x+--xx. -x+. -xx.*
xx--%x*-x+--x+--
%x*-x+**x+**+**
+**xx*-xx--x+-*
+x**xx**x+*x+**
xx**xx-*xx**x+*-
x+*xx*-+x+*xx--
+*+*+x*-xx**x+--

```

Sl. 3.6. Ispis termalne slike prilikom PRINT\_ASCIIART načina rada

```

26.2, 26.3, 26.3, 26.4, 26.3, 26.4, 26.3, 26.2, 25.9, 26.0, 26.0, 26.1, 26.0, 25.8, 25.7, 25.8,
26.2, 26.3, 26.5, 26.6, 27.0, 27.1, 26.5, 26.3, 26.1, 26.0, 25.9, 26.0, 26.1, 26.0, 26.0, 26.1,
26.3, 26.6, 27.0, 28.4, 29.8, 29.8, 28.6, 27.1, 26.6, 26.3, 26.1, 26.0, 26.2, 26.0, 26.2, 26.1,
26.1, 26.2, 26.4, 27.7, 28.8, 30.0, 30.0, 29.1, 28.6, 27.2, 26.2, 26.3, 26.3, 26.0, 26.1, 26.2,
26.2, 26.4, 26.8, 28.3, 29.7, 31.0, 31.1, 30.9, 30.0, 29.3, 28.6, 27.2, 26.4, 26.1, 26.1, 26.2,
26.0, 26.2, 26.3, 26.8, 28.6, 30.0, 31.2, 31.4, 31.3, 29.7, 28.6, 26.9, 26.5, 26.2, 26.0, 26.1,
25.8, 26.0, 26.2, 26.9, 28.2, 30.1, 31.2, 31.4, 31.5, 31.2, 31.1, 29.9, 27.3, 26.3, 26.2, 26.3,
25.8, 25.7, 26.0, 26.3, 27.3, 28.4, 30.1, 31.4, 31.4, 31.4, 31.4, 31.2, 28.5, 26.8, 26.1, 26.2,
25.7, 25.7, 26.3, 27.3, 28.3, 28.7, 30.4, 31.2, 31.5, 31.6, 31.5, 31.4, 29.9, 27.0, 26.3, 26.1,
25.8, 26.0, 26.3, 27.0, 27.1, 27.9, 28.1, 30.1, 31.4, 31.4, 31.5, 31.6, 31.4, 31.1, 26.0, 27.3,
26.2, 25.8, 26.2, 26.0, 26.5, 27.0, 27.9, 28.7, 30.0, 31.8, 31.9, 32.0, 32.0, 32.1, 31.9, 31.8,
25.8, 25.9, 25.8, 26.1, 26.1, 26.0, 27.1, 27.6, 28.9, 30.2, 31.8, 31.7, 32.1, 31.8, 32.0, 31.9,

```

Sl. 3.7. Ispis brojčanih vrijednosti temperatura prilikom PRINT\_TEMPERATURES načinu rada

Nakon svih potrebnih deklaracija kreće startup() funkcija Arduino programa. U njoj je prvo definiran pin LE diode alarmantnog stanja kao izlaz – na njega se signal šalje, tj. ne očekuju se podaci natrag prema mikroupravljaču s tog pina. Zatim se pokreće serijska komunikaciju. Pomoću nje i Arduino Serial Monitor-a prikazuju se temperature, odnosno ASCII simboli na zaslonu. Bitna stavka kod serijske komunikacije je baud rate – količina informacija koja prolazi kroz komunikacijski kanal. U kontekstu serijske komunikacije „115200 baud“ znači da kroz kanal serijske komunikacije prolazi maksimalno 115.200 bita po sekundi. Na brzinama većim od 76.800 bit/s potrebno je obratiti pažnju i na dužinu kabla za programiranje – veća brzina zahtjeva manji put u istom vremenu prijenosa podataka. [16] Nadalje, nakon uspostavljanja serijske komunikacije ispisuju se osnovni podaci o senzoru i postavljaju inicijalne vrijednosti, odabire se „interleaved“ način rada podržan od strane senzora i rezolucija ulaza. Rezolucija označava broj bitova koji se šalje prema i od mikroupravljača. Tako primjerice tvornička vrijednost od 18 bita rezolucije

označava  $2^{18}$  bajta podataka u toku. Potrebno je postaviti i stopu osvježavanja (engl. *refresh rate*), tj. koliko puta u sekundi senzor prikuplja nove podatke. U slučaju MLX90641 senzora vrijednosti stope osvježavanja kreću se od  $2^{-1}$  (0,5) Hz do  $2^6$  (64) Hz.

```
void setup() {  
  pinMode(pinToControl, OUTPUT);  
  Serial.begin(115200);  
  delay(100);  
}
```

Programski kod 3.4. Inicijalizacija serijske komunikacije

```
Serial.print("Serial number: ");  
Serial.print(mlx.serialNumber[0], HEX);  
Serial.print(mlx.serialNumber[1], HEX);  
Serial.println(mlx.serialNumber[2], HEX);  
  
mlx.setMode(MLX90641_INTERLEAVED);  
//mlx.setMode(MLX90641_CHESS);  
Serial.print("Current mode: Interleave ");  
/* if (mlx.getMode() == MLX90641_CHESS) {  
  | Serial.println("Chess");  
  } else {  
  | Serial.println("Interleave");  
  */  
}
```

Programski kod 3.5. Ispis osnovnih podataka senzora i postavljanje radnom načina

```
mlx.setResolution(MLX90641_ADC_18BIT);  
Serial.print("Current resolution: ");  
mlx90641_resolution_t res = mlx.getResolution();  
switch (res) {  
  | case MLX90641_ADC_16BIT: Serial.println("16 bit"); break;  
  | case MLX90641_ADC_17BIT: Serial.println("17 bit"); break;  
  | case MLX90641_ADC_18BIT: Serial.println("18 bit"); break;  
  | case MLX90641_ADC_19BIT: Serial.println("19 bit"); break;  
}
```

Programski kod 3.6. Postavljanje i ispisivanje korištene rezolucije



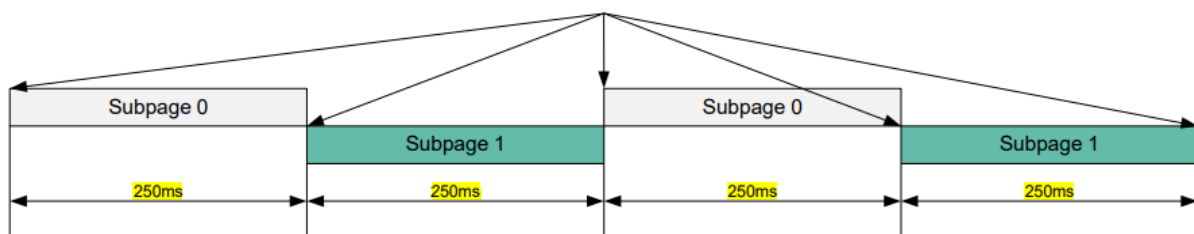
```

mlx.setRefreshRate(MLX90641_64_HZ);
Serial.print("Current frame rate: ");
mlx90641_refreshrate_t rate = mlx.getRefreshRate();
switch (rate) {
  case MLX90641_0_5_HZ: Serial.println("0.5 Hz"); break;
  case MLX90641_1_HZ: Serial.println("1 Hz"); break;
  case MLX90641_2_HZ: Serial.println("2 Hz"); break;
  case MLX90641_4_HZ: Serial.println("4 Hz"); break;
  case MLX90641_8_HZ: Serial.println("8 Hz"); break;
  case MLX90641_16_HZ: Serial.println("16 Hz"); break;
  case MLX90641_32_HZ: Serial.println("32 Hz"); break;
  case MLX90641_64_HZ: Serial.println("64 Hz"); break;
}

```

Programski kod 3.7. Postavljanje stope osvježavanja senzora te ispis

Nakon postavljanja početnih uvjeta, kreće radni dio programa, tj. loop() funkcija koja iznova izvodi isti programski kod i svakim prolaskom kroz kod dobiva nove podatke od senzora. U svrhu pravilnog dohvaćanja podataka sa senzora učitavaju se obje stranice podataka sa senzora nad kojima se vrše kalkulacije. Željena stopa osvježavanja je 64 Hz što znači da je za osvježavanje jedne stranice potrebno  $\frac{1}{64}$  sekundi. Taj broj pomnožen s dvije stranice daje približno 32 ms – iznos čekanja prije ponovnog dohvaćanja podataka s MLX90641 senzora.



Sl. 3.8. Osvježavanje stranica prilikom 4 Hz stope osvježavanja [2]

```

void loop() {
  count = 0;
  delay(32);
  if (mlx.getFrame(frame) != 0) {
    Serial.println("Failed");
    return;
  }
}

```

Programski kod 3.8. Postavljanje čekanja u funkciji loop() i dohvaćanje podataka

Korištenjem `.getFrame()` funkcije definirane unutar biblioteke dohvaćaju se podaci sa senzora i spremaju u lokalnu varijablu `frame`. Dopisana je i pripadajuća poruka u slučaju neuspješnog dohvaćanja podataka. U programskom kodu 3.8. spomenuta `count` varijabla koristi se za izbjegavanje pokretanja alarmantnog stanja prilikom pojave stršćih vrijednosti.

U drugom dijelu `loop()` funkcije formatira se ispis podataka. Pomoću dvije petlje jednodimenzionalno polje sa 192 mjesta ispisuje se u formatu dvodimenzionalnog polja sa 16 stupaca i 12 redaka. Varijabla `float t` predstavlja očitane temperaturu pojedinog piksela.

```
for (uint8_t h=0; h<12; h++) {  
    for (uint8_t w=0; w<16; w++) {  
        float t = frame[h*16 + w];
```

Programski kod 3.9. Formatiranje ispisa

Zadnji dio radne funkcije grana se pomoću direktiva. One predstavljaju uputu predprocesoru koji dio koda se uvažava, odnosno zanemaruje prilikom prevođenja programa. U ovom slučaju, provjerava se ranije definiran način rada – `PRINT_ASCIIART` ili `PRINT_TEMPERATURES`. Ispisivanje temperatura je jednostavnije budući da je to ono što senzor i daje. Koristi se funkcija `Serial.print()` koja preko Serial Monitora ispisuje brojčane vrijednosti na zaslon.

```
#ifdef PRINT_TEMPERATURES  
    Serial.print(t, 1);  
    Serial.print(", ");  
#endif  
#ifdef PRINT_ASCIIART  
    char c = '&';  
    if (t < 15) c = ' ';  
    else if (t < 20) c = '.';  
    else if (t < 25) c = '-';  
    else if (t < 30) c = '*';  
    else if (t < 35) c = '+';  
    else if (t < 40) c = 'x';  
    else if (t < 45) c = '%';  
    else if (t < 50) c = '#';  
    else if (t < 55) c = 'X';  
    else c = '0';  
    Serial.print(c);  
  
    if(t > 90) count = count + 1;  
#endif
```

Programski kod 3.10. Predprocesorske naredbe i ispisivanje vrijednosti

Drugi način rada zahtjeva uvjetno grananje. Korištenjem **if/else** vrijednost temperature zamjenjuje ASCII simbol radi lakše vizualizacije toplinske slike koju senzor daje. Ukoliko više piksela prepozna vrlo visoku temperaturu pali se alarmantno stanje.

Posljednji dio koda je alarmantno stanje. Sastoji se od jednog uvjetnog grananja koje glasi: ako više od 5 piksela prepozna vrlo visoku temperaturu, naizmjenično pali/gasi LE diodu – u suprotnom ju drži ugašenom. Dodana je i pauza u slučaju paljenja/gašenja LE diode kako bi ljudsko oko moglo prepoznati promjenu – zbog tromosti oka potrebno je imati pauzu programa dužu od 40 ms kako bi oko prepoznalo da se dioda ugasila, odnosno upalila [17].

```
if(count>5){
digitalWrite(pinToControl, HIGH);
delay(50);
digitalWrite(pinToControl, LOW);
}
else
digitalWrite(pinToControl, LOW);

Serial.println();
}
```

Programski kod 3.11. Upravljanje LE diodom u alarmantnom stanju

## 4. EKSPERIMENT

Kako bi se dokazala ispravnost mikroupravljačkog sustava odrađen je eksperiment. U prvom dijelu eksperimenta iznad senzora je postavljen hladan predmet. Za temperature niže od 15°C senzor šalje brojevanu vrijednost, a program na zaslon ispisuje prazan prostor.

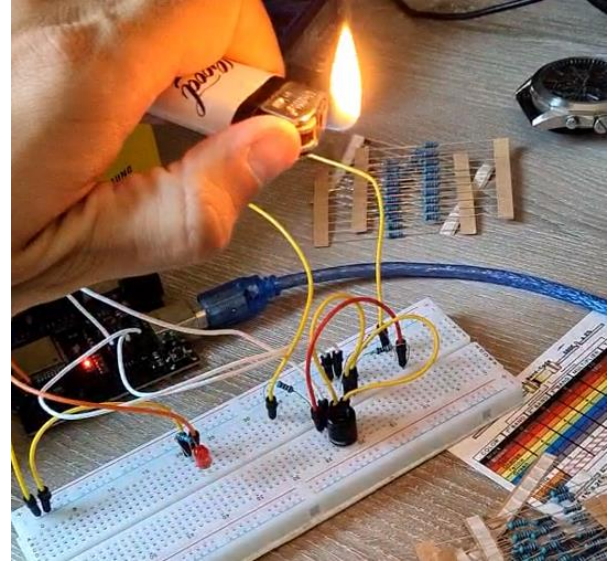
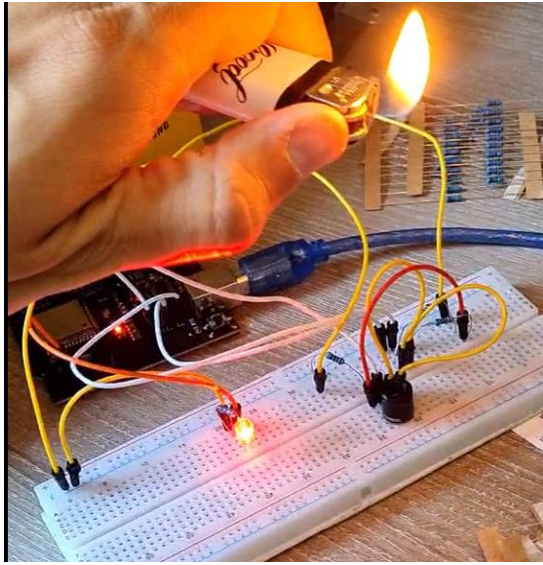


Sl. 4.1. Hladni predmet

XX+X+	++XXX++X++	XX+X++++	XXX+X+X++X+X+X
X++X	+XX+XXX+	X++XX+X+	XX+X+X++X+XXX+X+
X+	++X+X++	X+XX+X+X+	XX+X++X+X+X+X++X
X	+X+X+X	XXX+X+X++	XX+X+X++X+X+XXX+
X	XX+X++	XX+X+X++X+	+X+X+X+X+X +X+X
X	+X++X	XX+X++X+X+	+XX+X+X+ +X+
X	+XXX+	XX+X+X++X+X+	X+X+X++ +X
+	++X+X	+X+X+X+X+X+++	X+XX+X X
+X	+X++X+	+XX+X+X+X++X++	+X+XXX
X+	X++X+X	X+X+X++X+X++X+X	XX+X++
X+X	X+X++XX	X+XX+X+X+X+X++XX	X++XX+
+X+X	X+XX+X++	+X+XXXX+X+XX+X++	X+XX+XX

Sl. 4.2. Pomicanje hladnog predmeta po vidokrugu senzora i pomicanje prostora niske temperature na ispisu

U drugom dijelu eksperimenta korišten je upaljač kao izvor plamena te simulacija požara. Nakon što je plamen doveden u senzoru vidljivo područje, pali se „blicanje“ LE diode koja označava alarmantno stanje.



Sl. 4.3. Paljenje i gašenje LE diode tokom alarmantnog stanja

```

0000++++XXX++X++
0000x+x+x+xx+xxx
00000x+x+xxx+x+x
0000000++x+x+x+x
0000000+x+xx+x++
00000000x+x+x++x
00000000x+x+xxx+
000000+x+x++xx+x
000000+x++xx+xx+
00000++xx+xx+++x
000x+x+x+x+x++xx
0x+xxxxx+x+xx+x++

```

Sl. 4.4. Izlaz prilikom prilaska plamena senzoru s lijeve strane – simboli O označavaju visoke temperature

Cijeli video eksperimenta dostupan je na linku [18].

## 5. ZAKLJUČAK

Mikroupravljački sustav za prepoznavanje požara uspješno je napravljen. Omogućena je prevencija potencijalno opasnih situacija za čovjeka i za imovinu. MLX90641 senzor nalazi se u kompaktnom pakiranju te je cijelim sustavom lako rukovati i prenijeti ga. Moguća je i njegova ugradnja na mjesta bez pristupa internetu budući da se sva komunikacija događa interno – unutar samog senzora. Spajanjem uređaja na računalo moguće je bilježiti svaku sliku koja nastaje te ih spremati radi dokumentacije unutar Serial Monitora ugrađenog u Arduino IDE. Ovakvim mikroupravljačkim sustavima lako je rukovati te ih nadograditi, a uz pomoć strojnog učenja i osamostaliti. ESPduino-32 platforma za testiranje pokazala se kao odličan alat kako za početničke elektroničke projekte, tako i za napredne mikroupravljačke sustave. Prednost nad ostalim razvojnim platformama je velika količina dostupne memorije bez koje ovaj rad ne bi bio moguć – kroz rad otkriveno je da Arduino UNO nema dostatnu memoriju za izvođenje zadatka.

Moguća unaprjeđenja ovog mikroupravljačkog sučelja su spajanje senzora na mrežu te, ako mreža postoji, automatski pozvati u pomoć u slučaju otkrivanja požara. Spajanje mlaznice protupožarnog aparata bilo bi privremeno rješenje problema budući da mala rezolucija samog senzora neće biti dovoljna za precizne akcije gašenja.

## LITERATURA

- [1] Shedboy71, ESP32learning, A LOOK AT THE ESPDUINO-32 BORD, 12th November 2017, dostupno na [<http://www.esp32learning.com/hardware/a-look-at-the-espduino-32-board.php>] Pristupljeno dana 28.8.2023.
- [2] Mouser Electronics, MLX90641ESF-BCB-000-SP službena dokumentacija, dostupno na: <https://www.mouser.co.uk/ProductDetail/Melexis/MLX90641ESF-BCB-000-SP?qs=17cgNqFNU1j9Pjwrdu1MIg%3D%3D> Pristupljeno dana 1.6.2023.
- [3] electronics-lab, MLX90641 – FAR INFRARED THERMAL SENSOR ARRAY WITH 16X12PX RESOLUTION, dostupno na: <https://www.electronics-lab.com/mlx90641-far-infrared-thermal-sensor-array-16x12px-resolution/> Pristupljeno dana 1.6.2023.
- [4] Arduino Docs, Arduino službena web-stranica, dostupno na: <https://docs.arduino.cc/software/ide-v1/tutorials/arduino-ide-v1-basics> Pristupljeno dana 23.8.2023.
- [5] Scott Campbell , Circuit Basics, HOW TO INSTALL AND CONFIGURE THE ARDUINO IDE, dostupno na [<https://www.circuitbasics.com/wp-content/uploads/2020/05/Arduino-IDE-1-1024x878.png>] Pristupljeno dana 28.8.2023.
- [6] Visual Studio Code, 8.3.2023. dostupno na: URL [<https://code.visualstudio.com/docs/editor/whyvscode>] Pristupljeno dana 28.8.2023.
- [7] Barry, A Total Solution, Inc., FIRE PROTECTION: THE FOUR TYPES OF FIRE DETECTORS, 21.12.2017. dostupno na: URL [<https://www.atotalsolution.com/blog/fire-protection-the-four-types-of-fire-detectors/>] pristupljeno dana 31.8.2023.
- [8] Herschel, Frederick William. Hrvatska enciklopedija, mrežno izdanje. Leksikografski zavod Miroslav Krleža, 2021. dostupno na: <https://enciklopedija.hr/natuknica.aspx?ID=25231> Pristupljeno dana 25.8.2023.
- [9] infracrveno zračenje. Hrvatska enciklopedija, mrežno izdanje. Leksikografski zavod Miroslav Krleža, 2021. dostupno na: <https://enciklopedija.hr/natuknica.aspx?ID=27417> Pristupljeno dana 25.8.2023.
- [10] Wikipedia, Infrared, dostupno na: URL [[https://upload.wikimedia.org/wikipedia/commons/c/cf/Ir\\_girl.png](https://upload.wikimedia.org/wikipedia/commons/c/cf/Ir_girl.png)] Pristupljeno dana 31.8.2023.

[11] EESemi, dostupno na: URL [<https://www.eesemi.com/to39.htm>] Pristupljeno dana 25.8.2023.

[12] ] Scott Campbell , Circuit Basics, BASICS OF THE I2C COMMUNICATION PROTOCOL, dostupno na: URL [ <https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/> ] pristupljeno dana 31.8.2023.

[13] Robot Electronics, dostupno na: URL [ <https://www.robot-electronics.co.uk/images/i2ca.GIF> ] pristupljeno dana 31.8.2023.

[14] Goodness Woke, Redgate Hub, The difference between libraries and frameworks, 24.3.2023. dostupno na: URL [ <https://www.red-gate.com/simple-talk/development/other-development/the-difference-between-libraries-and-frameworks/#:~:text=A%20library%20is%20a%20collection,be%20called%20upon%20as%20needed.> ] Pristupljeno dana 26.8.2023

[15] Mouser Electronics, MLX90640 službena dokumentacija, dostupno na: <https://hr.mouser.com/c/ds/?marcom=145139989> Pristupljeno dana 28.8.2023.

[16] Nikko Phongchit, Setra, What is Baud Rate & Why is it important?, 4.1.2016. dostupno na: URL [ <https://www.setra.com/blog/what-is-baud-rate-and-what-cable-length-is-required-1> ] Pristupljeno dana 26.8.2023

[17] John V. Forrester, Andrew D. Dick, Paul G. McMenamin, Fiona Roberts, Eric Pearlman, The Eye, Chapter 5 Physiology of vision and the visual system, Elsevier, 2016

[18] [https://drive.google.com/file/d/1F7ZPYqR\\_jst2oBAkMNUwqWG\\_Uy85IsBh/view?usp=drive\\_link](https://drive.google.com/file/d/1F7ZPYqR_jst2oBAkMNUwqWG_Uy85IsBh/view?usp=drive_link)



## SAŽETAK

Tema ovog rada bila je upoznati se s tehnikama te skupiti alate, dizajnirati i programirati mikroupravljački sustav za otkrivanje požara. Potrebno je pomoću infracrvenog zračenja zapisati temperaturu okoline te u slučaju pojavljivanja visokih vrijednosti temperature obavijestiti korisnika. Za bilježenje temperatura korišten je MLX90641, infracrveni senzor topline a 16x12 matričnim prikazom. Sa 192 infracrvena senzora (piksela) u sebi može zabilježiti temperaturu od -40°C do 300°C čineći ga pogodnim za prikaz toplinskih snimaka. Pomoću Arduino IDE sučelja korisnik komunicira sa senzorom prikupljajući s njega podatke, a može svoj kod prepustiti zajednici programera u obliku gotovih biblioteka. Komunikacija se odvija preko I2C protokola, vrlo jednostavnog i materijalno nezahtjevnog načina. Koriste se dvije žice za napajanje te dvije za samu komunikaciju. Dostupne su velike brzine prijenosa što u prijevodu znači rad senzora u stvarnom vremenu.

Ključne riječi: ESPduino-32, MLX90641, Melexis, otkrivanje plamena, I2C, mikroupravljački sustav

## **ABSTRACT**

The topic of this work was to learn about the techniques and collect tools, design and program a microcontroller system for fire detection. It is necessary to record the ambient temperature using infrared radiation and inform the user if high temperature values appear. MLX90641, an infrared heat sensor with a 16x12 matrix display, was used to record said temperatures. With 192 infrared sensors (pixels), it can record temperatures from -40°C to 300°C, making it suitable for displaying thermal infrared images. Using the Arduino IDE interface, the user communicates with the sensor, collecting data from it, and can leave his code to the community in the form of ready-made libraries. Communication takes place via the I2C protocol, a very simple and materially undemanding method. Two wires are used for power and two for communication. High transfer rates are available, which translates into real-time sensor operation.

Keywords: ESPduino-32, MLX90641, Melexis, flame detection, I2C, microcontroller

## ŽIVOTOPIS

Matko Grgić rođen je 17.10.2001. u Osijeku. Pohađao je Osnovnu školu Jagode Truhelke u Osijeku i završio ju 2016. godine. Te godine upisao je III. gimnaziju Osijek po prirodoslovno-matematičkom programu. Za vrijeme pohađanja srednje škole nastupio je na brojnim školskim i županijskim natjecanjima iz područja Informatike. Nakon ispita državne mature, 2020. godine, upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku (FERIT) gdje odabire preddiplomski smjer računarstva i blok računalnog inženjerstva. Od 2022. godine član je IEEE inženjerske organizacije, a od 2023. je i član studentskog zbora istog fakulteta. Uz fakultetske obaveze, član je i odbojkaškog kluba MOK Marsonia iz Slavenskog Broda za koji profesionalno igra hrvatsku Superligu.

## PRILOZI

Kompletan programski kod mikroupravljačkog sustava za otkrivanje požara u Arduino IDE sučelju

```
#include <Adafruit_MLX90641.h>

Adafruit_MLX90641 mlx;
float frame[12*16]; // buffer for full frame of temperatures

// uncomment *one* of the below
//#define PRINT_TEMPERATURES
#define PRINT_ASCIIART
const int pinToControl = 26;
int count = 0;

void setup() {
  pinMode(pinToControl, OUTPUT);
  Serial.begin(115200);
  delay(100);

  Serial.println("MLX90641 Simple Test");
  if (! mlx.begin(MLX90641_I2CADDR_DEFAULT, &Wire)) {
    Serial.println("MLX90641 not found!");
    while (1) delay(10);
  }
  Serial.println("Found Adafruit MLX90641");

  Serial.print("Serial number: ");
  Serial.print(mlx.serialNumber[0], HEX);
  Serial.print(mlx.serialNumber[1], HEX);
  Serial.println(mlx.serialNumber[2], HEX);

  mlx.setMode(MLX90641_INTERLEAVED);
  //mlx.setMode(MLX90641_CHESS);
  Serial.print("Current mode: Interleave ");
  if (mlx.getMode() == MLX90641_CHESS) {
    Serial.println("Chess");
  } else {
    Serial.println("Interleave");
  }

  }

  mlx.setResolution(MLX90641_ADC_18BIT);
  Serial.print("Current resolution: ");
  mlx90641_resolution_t res = mlx.getResolution();
  switch (res) {
    case MLX90641_ADC_16BIT: Serial.println("16 bit"); break;
    case MLX90641_ADC_17BIT: Serial.println("17 bit"); break;
```

```

        case MLX90641_ADC_18BIT: Serial.println("18 bit"); break;
        case MLX90641_ADC_19BIT: Serial.println("19 bit"); break;
    }

    mlx.setRefreshRate(MLX90641_64_HZ);
    Serial.print("Current frame rate: ");
    mlx90641_refreshrate_t rate = mlx.getRefreshRate();
    switch (rate) {
        case MLX90641_0_5_HZ: Serial.println("0.5 Hz"); break;
        case MLX90641_1_HZ: Serial.println("1 Hz"); break;
        case MLX90641_2_HZ: Serial.println("2 Hz"); break;
        case MLX90641_4_HZ: Serial.println("4 Hz"); break;
        case MLX90641_8_HZ: Serial.println("8 Hz"); break;
        case MLX90641_16_HZ: Serial.println("16 Hz"); break;
        case MLX90641_32_HZ: Serial.println("32 Hz"); break;
        case MLX90641_64_HZ: Serial.println("64 Hz"); break;
    }
}

void loop() {
    count = 0;
    delay(32);
    if (mlx.getFrame(frame) != 0) {
        Serial.println("Failed");
        return;
    }
    Serial.println();
    Serial.println();

    for (uint8_t h=0; h<12; h++) {
        for (uint8_t w=0; w<16; w++) {
            float t = frame[h*16 + w];
#ifdef PRINT_TEMPERATURES
            Serial.print(t, 1);
            Serial.print(", ");
#endif
#ifdef PRINT_ASCIIART
            char c = '&';
            if (t < 15) c = ' ';
            else if (t < 20) c = '.';
            else if (t < 25) c = '-';
            else if (t < 30) c = '*';
            else if (t < 35) c = '+';
            else if (t < 40) c = 'x';
            else if (t < 45) c = '%';
            else if (t < 50) c = '#';
            else if (t < 55) c = 'X';
            else c = '0';
            Serial.print(c);

```

```
        if(t > 90) count = count + 1;
    #endif
    }

    if(count>5){
        digitalWrite(pinToControl, HIGH);
        delay(50);
        digitalWrite(pinToControl, LOW);
    }
    else
        digitalWrite(pinToControl, LOW);

    Serial.println();
}
}
```