

**NADOGRADNJA ZAVRŠNOG PROJEKTA:**  
**ROBOTSKA REKREACIJA RAČUNALNE IGRE PAC-MAN**

**Autor:** Viktorija Smlatić

**Mentor:** doc. dr. sc. Ana Sović Kržić

## **Sadržaj:**

<b>1. Uvod</b>	<b>3</b>
<b>2. Tehničke značajke VIDI Project X mikroračunala</b>	<b>4</b>
<b>3. Izrada projekta</b>	<b>5</b>
3.1 Infracrveni komunikacijski sustav i RC5 protokol	5
3.2 Programiranje odašiljača na VIDI Project X mikroračunalu	7
<b>4. Zaključak</b>	<b>11</b>
<b>5. Literatura</b>	<b>12</b>

## 1. Uvod

U završnom projektu smo izradili rekreaciju računalne igre "Pacman", uz pomoć 3 Thymio robota. Započeli smo s kreiranjem mape po kojoj će se roboti kretati (oblik FER loga), a zatim s programiranjem duhova i konačno Pac-mana. Duhovi su se gibalili na način da su pratili crne linije koje, kako su uvijek nesavršeno postavljene, pridonijele u nasumičnosti kretanju. Međusobno bi komunicirali i s Pac-manom, putem infracrvenih signala, kako bi ustanovili kraj igre. Pac-mana je kontrolirao igrač koristeći daljinski upravljač koji funkcionira po RC5 protokolu (najčešće Philipsov). Iako je projekt uspješno izrađen, možemo ga puno unaprijediti.

U ovom kratkom radu, u sklopu predmeta Seminar 1, pokušat ćemo unaprijediti kontroliranje samog Pac-mana, kako bi igra bila ugodnija. Daljinski upravljači su dosta nezgodni za igre, stoga smo na umu imali novi VIDI Project X mikroračunalo. Kao što vidimo na Slici 1, ono na sebi ima 6 gumbova te ekran, koji su idealni za programiranje igara. Cilj ovog projekta jest napraviti program na VIDI Project X mikroračunalu, koji će koristiti infracrvene zrake i priložene gumbove, da bi mogli kontrolirati Thymio robota.



*Slika 1 Izgled VIDI Project X mikroračunala*

## 2. Tehničke značajke VIDI Project X mikroračunala

VIDI Project X je razvojno mikroračunalo koje koristi ESP32 dvojezgreni procesor. Sadržava 8MB flash memorije i 4MB PSRAM i "touchscreen" ekran rezolucije 320 x 240 piksela. Uz to, postoje mnogo ulazno izlaznih jedinica, kako bi se cijela pločica mogla koristiti za različite svrhe. Osim standardnih ulaza i izlaza kao što sadrži Arduino Uno, ima implementiran odašiljač i prijemnik infracrvenih signala, 6 tipki, audio konektor, temperaturne senzore..itd. Sve što sadržava možete vidjeti na Slici 2.



Slika 2 Komponente VIDI Project X mikroračunala

<b>Proizvođač</b>	Espressif Systems
<b>Wi-Fi</b>	802.11 b/e/g/i/n, BT
<b>Frekvencija</b>	2.4 GHz do 2.5 GHz
<b>Brzina prijenosa podataka</b>	150 Mb/s
<b>Izlazna snaga</b>	20 dBm
<b>Security</b>	WPA/WPA2/WPA2 - Enterprise/WPS
<b>CPU</b>	Tensilica Xtensa Lx6 microprocessor @ 240 MHz
<b>Memorija</b>	520 KiB SRAM

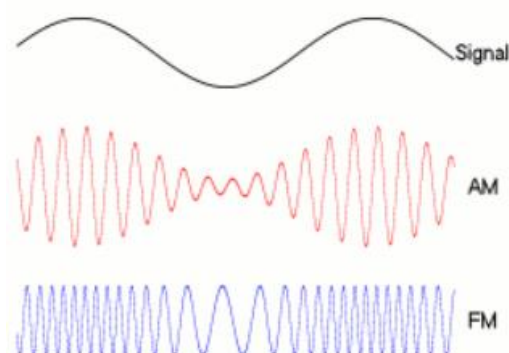
Tablica 1 Specifikacije mikroprocesora na VIDI Project X mikroračunalu

### 3. Izrada projekta

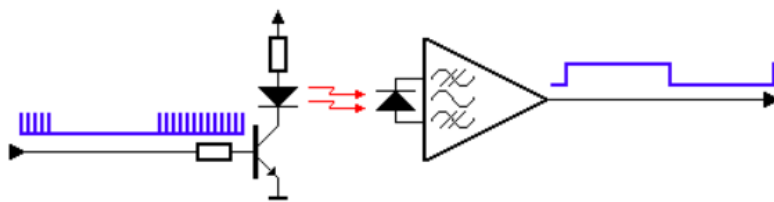
Znamo da za kontrolu Thymio robota daljinskim upravljačem, treba koristiti RC5 komunikacijski protokol. To znači da moramo slati određene infracrvene signale, kao što čini Philips daljinski upravljač. Za početak, objasniti ćemo kako takva komunikacija uopće radi.

#### 3.1 Infracrveni komunikacijski sustav i RC5 protokol

Da bi uspostavili komunikaciju putem infracrvenih signala, trebaju nam IR odašiljač, IR prijamnik i mikrokontroler. IR odašiljač izgleda kao obična ledica, samo što umjesto svijetla u vidljivom, kreira svjetlo u infracrvenom spektru. IR prijamnik je fotodioda koja pretvara infracrveno svjetlo u električni signal. Budući da oko nas postoji puno IR svjetlosnog šuma (sunce, žarulje...), koristi se tehnika modulacije signala. Koder na daljinskom upravljaču pretvara binarni signal u modulirani električni signal. To znači da se jedna veličina tog signala mijenja podudarno s podacima koje treba prenijeti (Slika 3). Na taj način smanjujemo utjecaj šuma okoline, jer se originalni sinusni oblik, koji koristimo, lagano izgubi u tom šumu. Nadalje, takav električni signal se pretvara u modulirani IR svjetlosni signal koji putuje do prijamnika. On ga, zatim, demodulira i dekodira u binarni signal, kojeg onda pošalje mikrokontroleru. Modulirani IR signal je niz infracrvenih svjetlosnih impulsa koji se uključuju i isključuju na visokoj frekvenciji koja je nama poznata. Ona se naziva nosećom frekvencijom (carrier frequency) te je glavna karakteristika signala po kojoj ga prijemnik razlikuje od šuma. Najčešće je ta frekvencija iznosa 38 kHz, a za RC5 protokol iznosi 36 kHz.



*Slika 3 Amplitudna i Frekvencijska modulacija sinusnog signala*

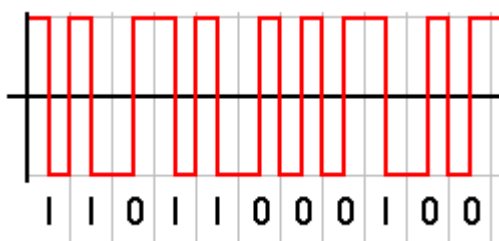


*Slika 4 Prikaz komunikacije putem infracrvenog svjetla*

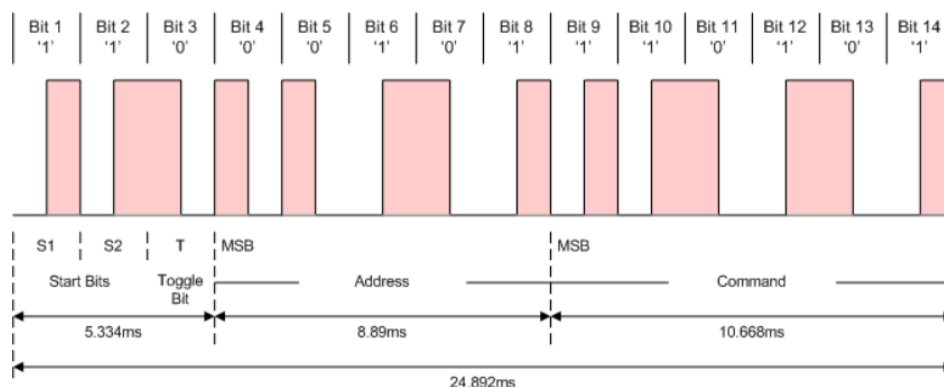
Uzorak binarnog signala koji mikrokontroler dobije je definiran protokolom. Postoji mnogo protokola: Sony, NEC i RC5 su najpopularniji. Svaki put kada pritisnemo tipku na daljinskom upravljaču, generira se jedinstveni heksadecimalni kod. Njega šalje odašiljač do prijmnika, koji ga prosljeđuje mikrokontroleru. Razlog zašto se generira jedinstveni, a ne nasumični kod je što ga mikrokontroler mora unaprijed poznavati kako bi dekodirao koja je tipka pritisnuta. Različiti upravljači i protokoli šalju različite kodove, stoga moramo otkriti koji kodovi su nama potrebni.

Objasnit ćemo RC5 protokol, budući da ga koristimo u projektu. Ovaj protokol koristi Manchester kodiranje, koji je prikazan na Slici 5. Za prijenos koda potrebno je 24,892 ms na frekvenciji 36 kHz. Kada se pritisne tipka, generira se poruka od 14 bita na sljedeći način:

- prva dva bita su '1'
- 3. bit je "toggle" - invertira se svaki put kada se tipka otpusti i ponovno pritisne (da znamo držimo li tipku ili pritišćemo više puta)
- sljedećih 5 bitova označava adresu za prijemnik
- ostalih 6 bitova predstavlja naredbu



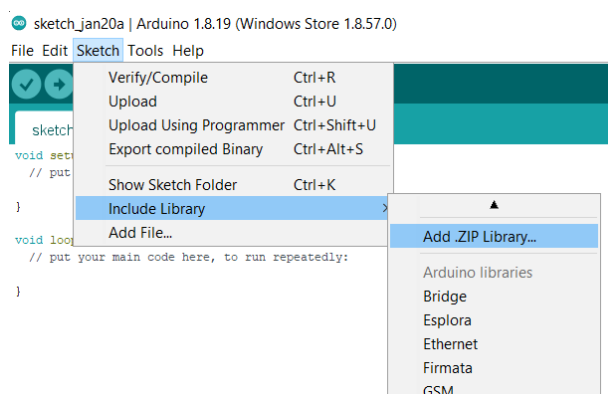
*Slika 5 Manchester kodiranje binarnog niza: 11011000100*



Slika 6 Primjer poruke RC5 protokola

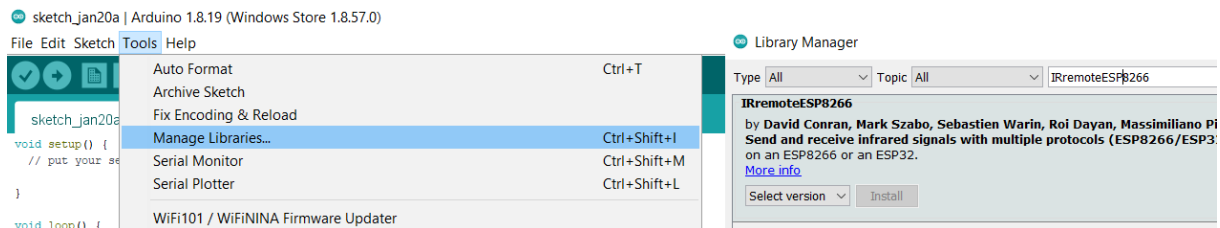
### 3.2 Programiranje odašiljača na VIDI Project X mikroračunalu

VIDI Project X mikroračunalo ima već ugrađen IR prijamnik i IR odašiljač, stoga nije potrebno dodatno spajati druge elektroničke komponente. Prije programiranja, potrebno je instalirati Arduino IDE i biblioteku IRremote. Arduino IDE ima detaljne upute za instalaciju na njihovoj web stranici, stoga to nećemo opisati ovdje. Biblioteku se može instalirati na dva načina. Prvi jest preuzimanjem .zip datoteke. Zatim u aplikaciji, idite na Sketch -> Include Library -> Add.ZIP Library... , kao što je prikazano na Slici 7.



Slika 7 Instalacija biblioteke preuzimanjem njene .zip datoteke

Drugi način je putem same aplikacije. Idite na Tools -> Manage Libraries... , nakon čega bi se trebao otvoriti prozor Manage Libraries. U tražilici utipikajte "IRremote8266" i pronađite istoimeni naslov te kliknite Install., kao na Slici 8.



*Slika 8 Instalacija biblioteke putem Arduino IDE aplikacije*

Sada, osim funkcija za programiranje, imamo već gotove primjere kodova koje možemo koristiti. Često projekti koriste ovakav način programiranja, zbog jednostavnosti. Za početak, trebamo pronaći na koje tipke reagira Thymio. Želimo upravljati Thymio robota, stoga moramo prvo pronaći tipke koji rade na Thymiu. Ako ga uključite i postavite u ljubičasti (poslušni) način rada, uperite Philipsov daljinski upravljač, Thymio će reagirati. On će se početi gibati na tipke 2,4,6,8 ili strelice. Za ostale tipke, samo će zasvijetliti crvenom ledicom u znak uspješne komunikacije. Uzet ćemo tipke brojeva za repliciranje. Sada kada znamo na koje tipke Thymio reagira, moramo otkriti koji su njihovi kodovi. Možemo na internetu pronaći informaciju, no ona nam nije dovoljna. Koristit ćemo već gotov primjer. Za njegovo otvaranje kliknite File -> Examples -> IRremoteESP8266 -> IRrecvDumpV2. Prije pokretanja koda, u 38. retku treba promijeniti vrijednost varijable "kRecvPin" na 25, koja označava poziciju prijemnika. Zatim uključite VIDI X, koji ste spojili na računalo putem USB kabela i zatim pritisnite strelicu udesno u gornjem lijevom kutku. Na Tools -> Serial Monitor pokrenite prozor i na daljinskom upravljaču pritisnite prethodno navedene tipke. Na ekranu će se pojaviti informacije o toj tipki. Zasada, samo ćemo kopirati te podatke u neku .txt datoteku sa strane.

```
13:39:11.961 -> uint32_t address = 0x10;
13:39:11.961 -> uint32_t command = 0x17;
13:39:11.961 -> uint64_t data = 0x808E817;
13:39:11.961 ->
13:39:11.961 ->
13:39:15.467 -> Timestamp : 000035.172
13:39:15.467 -> Library : v2.7.3
13:39:15.467 ->
13:39:15.467 -> Protocol : SAMSUNG
13:39:15.467 -> Code : 0x8086897 (32 Bits)
13:39:15.467 -> uint16_t rawData[67] = {4542, 4424, 610, 552,
582, 524, 582, 550, 582, 524, 610, 1602, 638, 524, 610,
498, 608, 524, 610, 498, 610, 524, 608, 498, 610, 524,
608, 1606, 634, 498, 608, 526, 604, 502, 580, 554, 576,
1662, 578, 1664, 578, 530, 576, 1664, 576, 556, 576, 530,
576, 558, 576, 1664, 576, 530, 576, 558, 576, 1664, 576,
530, 576, 1664, 576, 1664, 576, 1638, 578}; // SAMSUNG
```

*Slika 9 Primjer rezultata programa IRrecvDumpV2 korištenjem Samsung daljinskog upravljača*



Sada želimo poslati signal Thymio robotu. Koristit ćemo novi gotov kod, File -> Examples -> IRremoteESP8266 -> IRsendDemo . U ovom kodu moramo učiniti više izmjena. U 35. retku, "kIrLed" promijeniti na 15; pin gdje se nalazi naš odašiljač. U varijabli "rawData" spremamo onaj dugi niz pod istim imenom iz .txt datoteke koje smo spremili te mijenjamo broj iz uglatih zagrada u 23. U "loop()" dijelu, treba promijeniti funkciju "sendSony" u "sendRC5", te postaviti sljedeće vrijednosti u zagradama: 0x2, 14, 3. Prvi broj označava naredbu (kod tipke), drugi količinu bitova koju taj kod, odnosno poruka zahtjeva i zadnji broj ponavljanja. U pozivu funkcije "sendRaw", potrebno je u zagradama zapisati sljedeće: rawData, 23, 36. Promijenili smo noseću frekvenciju iz 38kHz u 36kHz, te promijenili koliko se podataka nalazi u varijabli "rawData". Konačno, uključimo VIDI X, pritisnite na strelicu u gornjem lijevom kutu i postavite Thymia u ljubičastom (poslušnom) načinu rada.

```
#include <Arduino.h>
#include <IRremoteESP8266.h>
#include <IRsend.h>

const uint16_t kIrLed = 15; // ESP8266 GPIO pin to use. Recommended: 4 (D2).

IRsend irsend(kIrLed); // Set the GPIO to be used to sending the message.

// Example of data captured by IRrecvDumpV2.ino
uint16_t rawData[23] = {914, 812, 1784, 808, 914, 814, 914, 812, 916, 812, 914, 814, 914, 814, 912, 814, 914, 816, 912, 814, 914, 1650, 1782};
// Example Samsung A/C state captured from IRrecvDumpV2.ino
/*uint8_t samsungState[kSamsungAcStateLength] = {
    0x02, 0x92, 0x0F, 0x00, 0x00, 0x00, 0xF0,
    0x01, 0xE2, 0xFE, 0x71, 0x40, 0x11, 0xF0};*/

void setup() {
    irsend.begin();
    #if ESP8266
        Serial.begin(115200, SERIAL_8N1, SERIAL_TX_ONLY);
    #else // ESP8266
        Serial.begin(115200, SERIAL_8N1);
    #endif // ESP8266
}

void loop() {

    Serial.println("Philips");
    irsend.sendRC5(0x2, 14, 5);
    delay(2000);
    Serial.println("a rawData capture from IRrecvDumpV2");
    irsend.sendRaw(rawData, 23, 36);
    delay(2000);
    Serial.println("a Samsung A/C state from IRrecvDumpV2");
    irsend.sendSamsungAC(samsungState);
    delay(2000);
}
```

*Slika 10 Izgled koda IRsendDemo nakon izmjene*

Nakon pokretanja koda, Thymio se počne kretati. Primijetimo da se ne ubrzava i ne pokreće opet ako ga ručno zaustavimo. Nakon duljeg promatranja, možemo uočiti da Thymio ne sluša svaki put naredbu, iako se ona konstantno šalje. Naime, ova biblioteka nije dovoljno dobro kodirala RC5 protokol te nas prijemnik nije dobro primio kodove. Naredba je točna, ali onaj niz brojeva nije. Razlog tomu je što ovaj prijemnik, kao već ugrađen, nije dobar za filtriranje šuma. Također, kada tako konstatno šaljemo isti kod, pretpostavljam da dođe do interferencije koje kasnije Thymio ne može razjasniti.

Sada ćemo probati ubaciti dio koda koji će omogućiti da nakon pritiskom gornje tipke, VIDI X pošalje kod. Na početku programa ubacit ćemo dvije linije koda:

```
int PinTipkalo = 35;
```

```
int StanjeTipkala;
```

Ovaj dio koda inicijalizira tipku "gore" i varijablu kojom ćemo znati je li pritisnuta. Zatim, u funkciji "void setup()", na početku, dodajemo sljedeći dio koda:

```
pinMode (PinTipkalo, INPUT_PULLUP);
```

Sada smo pripremili tipku. Konačno, u funkciji "void loop()" pišemo sljedeće:

```
StanjeTipkala = analogRead(PinTipkalo);
```

```
if (StanjeTipkala > 4000) {
```

Napomena: nakon zagrade dolazi ostatak koda koji smo prethodno pripremili te vitičasta zagrada. Ovim dijelom koda smo provjerili je li tipka pritisnuta i ako je, poslat će se signal. Nakon pokretanja programa, Thymio se skoro svaki put pokrene. No, primijetimo da postoji zakašnjenje od par sekundi, što nam nije idealno za nekakvu igricu. Možemo smanjiti vrijednost u "delay", no opet treba previše vremena.

```
#include <Arduino.h>
#include <IRremoteESP8266.h>
#include <IRsend.h>

const uint16_t kIrLed = 15; // ESP8266 GPIO pin to use. Recommended: 4 (D2).

IRsend irsend(kIrLed); // Set the GPIO to be used to sending the message.
int PinTipkalo = 35;
int StanjeTipkala;

// Example of data captured by IRrecvDumpV2.ino
uint16_t rawData[23] = {914, 812, 1784, 808, 914, 814, 914, 812, 916, 812, 914, 814, 914, 814, 912, 814, 914, 816, 912, 814, 914, 1650, 1782};
// Example Samsung A/C state captured from IRrecvDumpV2.ino
/*uint8_t samsungState[kSamsungAcStateLength] = {
  0x02, 0x92, 0x0F, 0x00, 0x00, 0x00, 0xF0,
  0x01, 0xE2, 0xFE, 0x71, 0x40, 0x11, 0xF0};*/

void setup() {
  irsend.begin();
  #if ESP8266
    Serial.begin(115200, SERIAL_8N1, SERIAL_TX_ONLY);
  #else // ESP8266
    Serial.begin(115200, SERIAL_8N1);
  #endif // ESP8266
  pinMode (PinTipkalo, INPUT_PULLUP);
}

void loop() {
  StanjeTipkala = analogRead(PinTipkalo);
  if (StanjeTipkala > 4000) {
    Serial.println("Philips");
    irsend.sendRC5(0x2, 14, 5);
    delay(2000);
    Serial.println("a rawData capture from IRrecvDumpV2");
    irsend.sendRaw(rawData, 23, 36);
    delay(2000);
    Serial.println("a Samsung A/C state from IRrecvDumpV2");
    irsend.sendSamsungAC(samsungState);
    delay(2000);
  }
}
```

*Slika 11 Izgled koda IRsendDemo nakon druge izmjene*

## 4. Zaključak

U ovom kratkom projektu pokušali smo nadograditi završni projekt. Umjesto da Pacmana upravljamo običnim daljinskim upravljačem, želimo ga upravljati VIDI Project X mikroračunalom. Pojasnili smo kako takva komunikacija putem infracrvenog svjetla radi te u Arduino IDE napisali kratak kod koji to realizira.

Rezultati iako naizgled uspješni, nisu zadovoljavajući. Thymio sporo reagira na naredbe te nije zagarantirano da će svaki put reagirati. Nakon mnogo istraživanja i testiranja, nisam uspjela naći rješenje problema. Potrebno je više znanja i iskustva u elektronici te dublje istraživanje.

## 5. Literatura

1. VIDI Project X:

<https://vidi-x.org/>

2. Kako postaviti daljinski upravljač i prijemnik na Arduinu:

<https://www.circuitbasics.com/arduino-ir-remote-receiver-tutorial/>

3. Philips RC5 infracrveni protokol prijenosa:

<https://techdocs.altium.com/display/FPGA/Philips+RC5+Infrared+Transmission+Protocol>

4. IR komunikacija:

<https://learn.sparkfun.com/tutorials/ir-communication/all>

5. Manchester kodiranje:

[https://hr.wikipedia.org/wiki/Manchester\\_kodiranje](https://hr.wikipedia.org/wiki/Manchester_kodiranje)