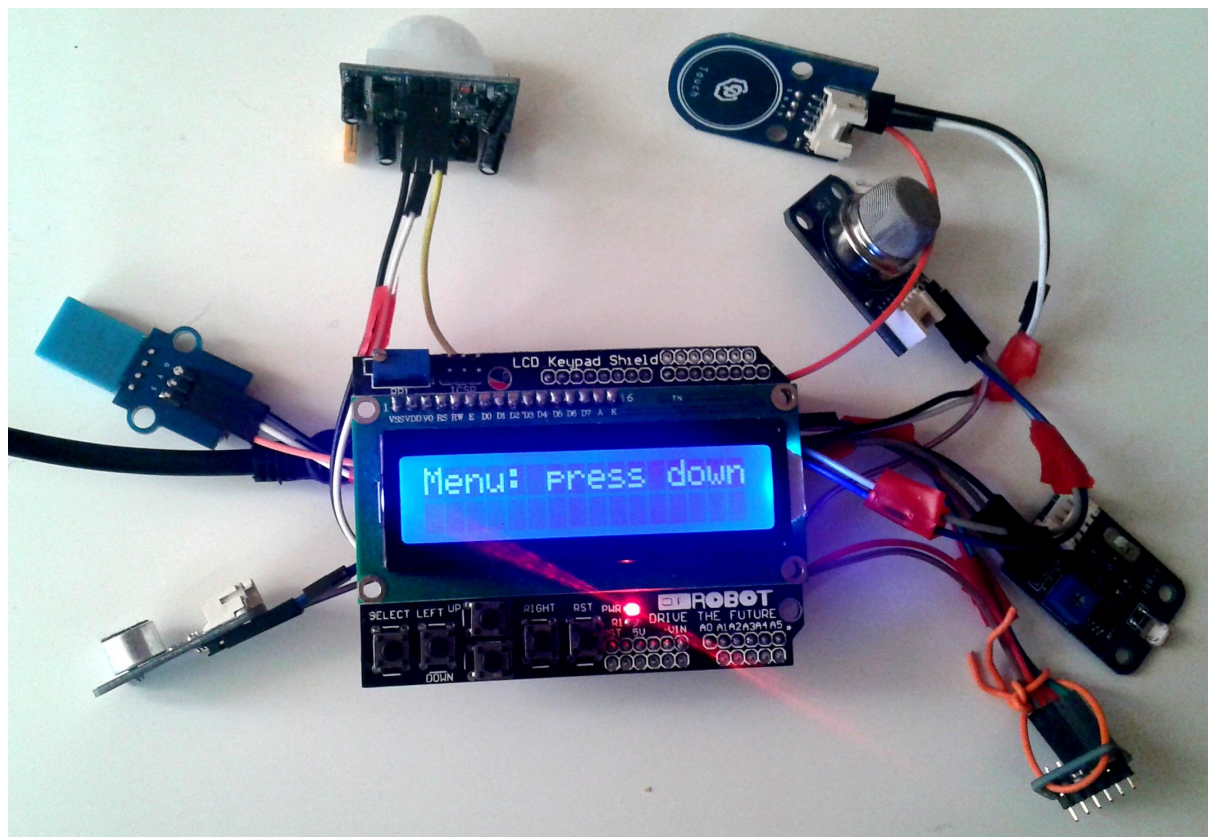


SENSOTRON



Kit za učenje o sedam osnovnih senzora:

1. Elektronski blok – senzor vlažnosti i temperature
2. Elektronski blok – senzor dodira/prekidač
3. Elektronski blok – zvučni senzor i mikrofon
4. Elektronski blok – svjetlosni senzor
5. MPL115A2 – I2C senzor atmosferskog pritiska/temperature
6. Elektronski blok – MQ-2 senzor plinova
7. PIR senzorski modul

Sadržaj

UVOD.....	3
Logika izbornika.....	5
Memorija.....	5
Shema spajanja.....	7
Dodane biblioteke.....	11
Elektronski blok – senzor vlažnosti i temperature [Nabava].....	11
Elektronski blok – senzor dodira/prekidač [Nabava].....	19
Elektronski blok – zvučni senzor i mikrofoni [Nabava].....	21
Elektronski blok – svjetlosni senzor [Nabava].....	22
MPL115A2 – I2C senzor atmosferskog pritiska/temperature [Nabava].	25
Elektronski blok – MQ-2 senzor plinova [Nabava].....	30
PIR senzorski modul [Nabava].....	32

Ovaj dokument slobodno možete:

Dijelite dalje — možete umnažati i redistribuirati materijal u bilo kojem mediju ili formatu

Stvarajte prerade — možete remiksirati, mijenjati i prerađivati djelo

Pod sljedećim uvjetima:

Imenovanje — Morate [adekvatno navesti autora](#), uvrstiti link na licencu i [naznačiti eventualne izmjene](#). Možete to učiniti na bilo koji razuman način, ali ne smijete sugerirati da davatelj licence izravno podupire Vas ili Vaše korištenje djela.

Nekomercijalno — Ne smijete koristiti materijal u [komercijalne svrhe](#).

Dijeli pod istim uvjetima — Ako remiksirate, mijenjate ili prerađujete materijal, Vaše prerade morate distribuirati pod [istom licencom](#) pod kojom je bio izvornik.

Bez daljnjih ograničenja — Ne smijete dodavati pravne uvjete ili [tehničke mjere zaštite](#) koji će druge pravno ograničiti da čine ono što im licenca dopušta.

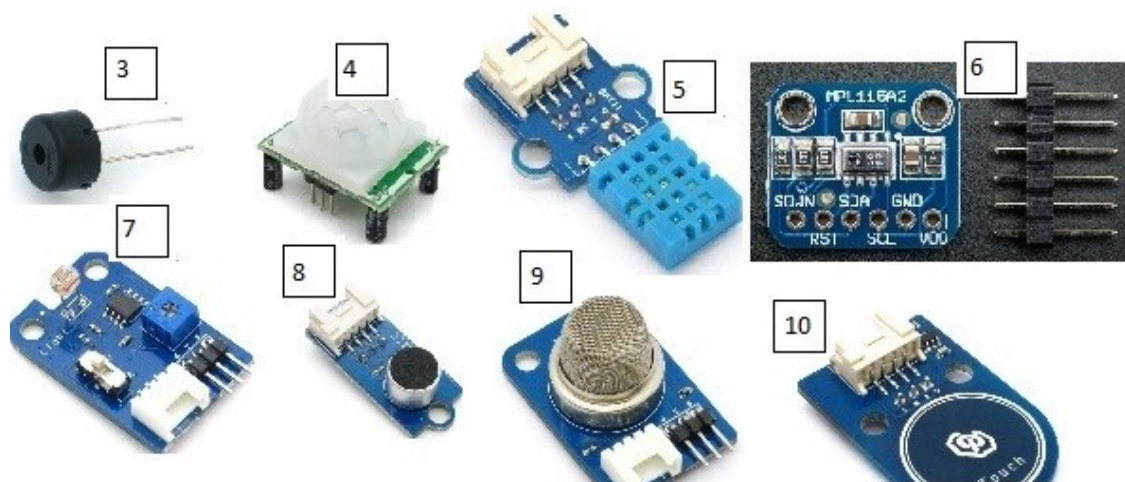
Puni tekst licence: <http://creativecommons.org/licenses/by-nc-sa/3.0/hr/legalcode>

UVOD

Poglavlja su podijeljena po nazivima senzora. U svakom dijelu biti će navedeno čemu senzor služi, kako je u ovom kitu spojen, kako je isprogramiran i koji su se problemi pojavili pri spajanju i programiranju te koje je pojedinosti potrebno znati za početnike koji bi na tom dijelu mogli zapeti.

Senzori (elektronski blokovi: senzor za vlažnost i temperaturu[5], senzor dodira/prekidač[10], zvučni senzor i mikrofون[8], svjetlosni senzor[7], MQ-2 senzor plinova[9], te I2C senzor atmosferskog pritiska i temperature MPL115A2[6], PIR senzorski modul[4] i PS1240 Piezo Buzzer[3]) se spajaju na mikrokontroler pločice Iteaduina Una[2] jumper ženskim žicama, a LCD Arduino štit (DFR0009)[1] ima već ugrađene pinove tako da se samo 'natakne' na pločicu. Sve komponente su pribavljene u webshopu mikrotron.hr, a direktni linkovi pojedinačne nabave navedeni su uz naslov senzora.





Kako bi se na LCD-u 1602 mogli prikazati svi senzori morao se omogućiti izbornik putem ugrađenih gumbiju na LCD štitu. Primjer kako izgleda ponuđeni senzor na LCD zaslonu [[Nabava](#)]:



Primjer kako izgleda ispis očitavanja temperaturnog senzora:



Logika izbornika

S obzirom na položaj gumbića odnosno izbornik

MENI



logika je sljedeća: prvo se koristi 'Down' kako bi se išlo niz izbornik, 'Up' kako bi se vratilo gore po želji, 'Select' uvijek izvršava rad odnosno test senzora i prikazuje rezultat, 'Right' ako se želi vratiti u meni dok 'Reset' vraća izbornik u prvotno stanje.

Memorija

Glede memorije, postoje tri memorije u mikrokontroleru na AVR Arduino pločama: Flash memorija u koju se učitava Arduino sketch, u SRAM(static random access memory)-u se sketch kreira i ovdje se manipulira s varijablama pri izvođenju, a EEPROM je memorija koju programeri mogu koristiti kako bi pohranili bazične informacije.

Informacije u SRAM-u se gube s gubitkom napajanja. Za primjer, na ovoj ploči ima za Flash: 32kB, SRAM 2kB, a EEPROM 1kB. [[Memorija](#)].

Na optimizaciju dakle treba paziti pa su tako jako bitni optimalni odabiri tipova podataka, izbjegavanje korištenja String strukture i raznih funkcija poput `clear()` LCD-a, `digitalWrite()` čiju ćete zamjenu vidjeti u kodu, a na ovoj stranici je objašnjena kao i niz drugih dobrih stvari [[Više](#)].

Bazično, trebalo bi se držati dobrog starog 'niskog' C-a. S obzirom na vrste memorija a količinu teksta odnosno znakova koji tu memoriju pune do mjere da se naredbe više ne izvršavaju kako treba i dolazi do raznih grešaka u izvedbi bilo je nužno koristiti PROGMEM sintaksu. Pritom je potrebno upozoriti sve koji će se 'polakomiti' za gotovim klasama odnosno pojednostavljenom (sintaktičkom) upotrebom PROGMEM-a. Npr. internetom kruži 'Flash.h' [[Flash](#)] i praktički se mistificira PROGMEM - implicira se (po internetu) kako je strašno teško koristiti izvornu sintaksu i da zapravo odmah treba prijeći na ponuđeno 'upakirano' rješenje. Nakon bezbroj problema glede kompajliranja koda pri čemu su greške bile navodno vezane za verziju IDE-a, odlučili smo se vratiti na početak (kako i treba kada se stvari počnu komplicirati) i odgovor je bio jasan i jednostavan.

[PROGMEM](#)

Eto to se dešava kada unaprijed napravite 'bauk' od nečega. Primjer upotrebe PROGMEM-a u ovom kodu:

```
#include <avr/pgmspace.h>

const char msgList0[] PROGMEM = "Menu: press down";
...
const char* const msgList[] PROGMEM = {msgList0,...,msgList11};

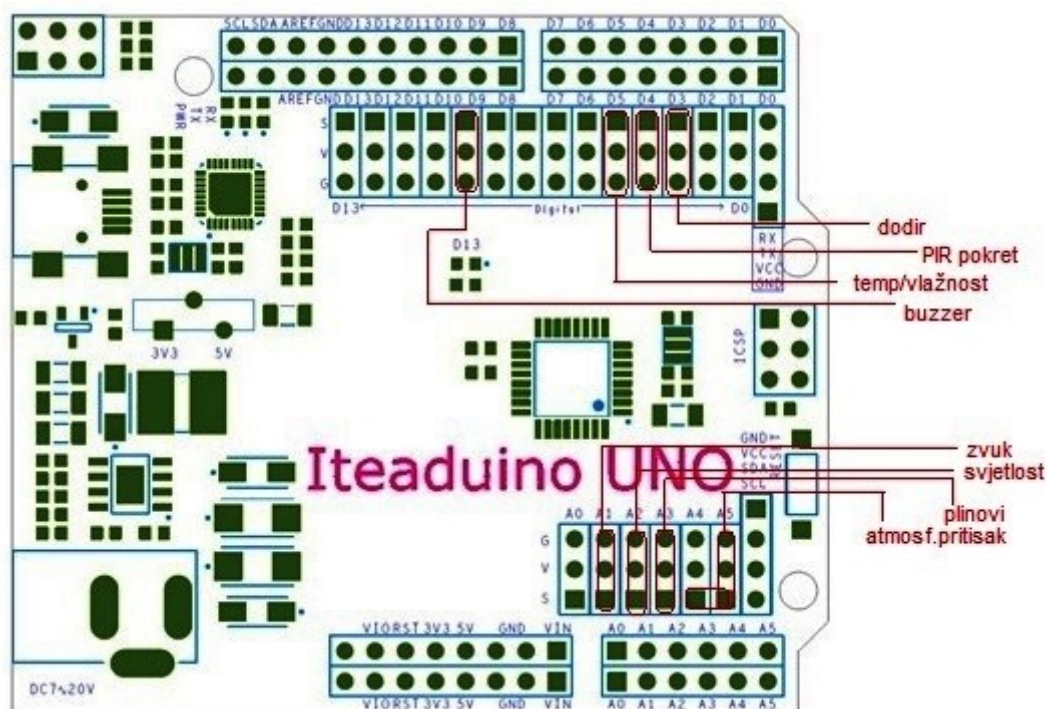
void setup()
{
    ...

    strcpy_P(buffer, (char*)pgm_read_word(&(msgList[0])));

    // prenesi buffer u funkciju za ispis na LCD-u
    lcdBasic(0, 0, buffer, 500);
}
```

Shema spajanja

Sada ćemo prikazati shemu spajanja, ne pravu tehničku nego laičku kako bi svi nepogrešivo shvatili gdje je točno koji senzor spojen i kako se ne bi zasebno u poglavljima opterećivao dokument pojedinačnim slikama [modificirano sa [Wikiltead](#)]:



Također potrebno je obratiti pozornost na parametre konstruktora LiquidCrystal-a s obzirom da se olako shvaćaju kada se počinje raditi s LCD-om. Slijedi 'Hello World' testni kod koji je jedan od mnogih koji ne radi iz istog razloga:

```
/*
```

```
  LiquidCrystal Library - Hello World
```

```
  Demonstrates the use a 16x2 LCD display.  The LiquidCrystal
  library works with all LCD displays that are compatible with the
```

```
  Hitachi HD44780 driver. There are many of them out there, and
  you
```

```
  can usually tell them by the 16-pin interface.
```

```
  This sketch prints "Hello World!" to the LCD
  and shows the time.
```

The circuit:

- * LCD RS pin to digital pin 12
- * LCD Enable pin to digital pin 11
- * LCD D4 pin to digital pin 5
- * LCD D5 pin to digital pin 4
- * LCD D6 pin to digital pin 3
- * LCD D7 pin to digital pin 2
- * LCD R/W pin to ground
- * 10K resistor:
 - * ends to +5V and ground
- * wiper to LCD VO pin (pin 3)

Library originally added 18 Apr 2008

by David A. Mellis

library modified 5 Jul 2009

by Limor Fried (<http://www.ladyada.net>)

example added 9 Jul 2009

by Tom Igoe

modified 22 Nov 2010

by Tom Igoe

This example code is in the public domain.

<http://www.arduino.cc/en/Tutorial/LiquidCrystal>

*/

```
// include the library code:
```

```
#include <LiquidCrystal.h>
```

```
// initialize the library with the numbers of the interface  
pins
```

```
LiquidCrystal lcd(3, 4, 5, 6, 7, 8);
```

```
void setup() {
```

```
  // set up the LCD's number of columns and rows:
```

```
  lcd.begin(16, 2);
```

```
  // Print a message to the LCD.
```

```
  lcd.print("hello, world!");
```

```
}
```

```
void loop() {
```

```
  // set the cursor to column 0, line 1
```

```
  // (note: line 1 is the second row, since counting begins  
with 0):
```

```
  lcd.setCursor(0, 1);
```

```
  // print the number of seconds since reset:
```

```
  lcd.print(millis()/1000);
```



```
}
```

Problem je u funkciji

```
LiquidCrystal lcd(3, 4, 5, 6, 7, 8);
```

čiji su ispravni parametri bili

```
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
```

Jer to je način na koji je proizvođač odabrao rasporediti žice na LCD štitu (ali to ne znači da pinovi ne mogu biti ožičeni drugačije):

```
LiquidCrystal(rs, enable, d4, d5, d6, d7)
```

Sada će LCD raditi i ispisivati 'Hello World'. Također treba napomenuti da se pritisnuti gumbi na LCD-u čitaju sa analognog pina A0 pa bi bilo poželjno da ostane slobodan, a D10 digitalni je za pozadinsko svjetlo na lcd-u itd., oko toga se treba informirati kako ne bi bilo kolizija.

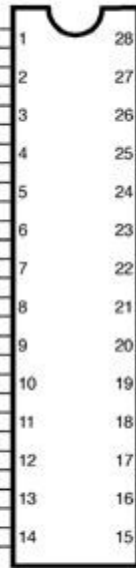
Još ćemo ukratko objasniti svrhu elektronskog bloka koji je poželjan za početnike. Poželjan je jer je standardiziran čime se ostvaruje lako spajanje, praktički kao da se lego kockice slažu. Kod rada sa svim senzorima uglavnom treba paziti na iste stvari kako se nebi oštetili, ali

najviše na temperaturu i izloženost svjetlosti te kemikalijama u radnoj okolini, što treba proučiti detaljnije izvan ovog dokumenta zasebno o svakom.

Jedna bitna slika 'Istina o pinovima' koja će početnicima mnogo olakšati put [[TruePins](#)]:

Atmega168 Pin Mapping
Arduino function

reset	(PCINT14/RESET) PC6	1
digital pin 0 (RX)	(PCINT16/RXD) PD0	2
digital pin 1 (TX)	(PCINT17/TXD) PD1	3
digital pin 2	(PCINT18/INT0) PD2	4
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5
digital pin 4	(PCINT20/XCK/T0) PD4	6
VCC	VCC	7
GND	GND	8
crystal	(PCINT6/XTAL1/TOSC1) PB6	9
crystal	(PCINT7/XTAL2/TOSC2) PB7	10
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12
digital pin 7	(PCINT23/AIN1) PD7	13
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14



28	PC5 (ADC5/SCL/PCINT13)
27	PC4 (ADC4/SDA/PCINT12)
26	PC3 (ADC3/PCINT11)
25	PC2 (ADC2/PCINT10)
24	PC1 (ADC1/PCINT9)
23	PC0 (ADC0/PCINT8)
22	GND
21	AREF
20	AVCC
19	PB5 (SCK/PCINT5)
18	PB4 (MISO/PCINT4)
17	PB3 (MOSI/OC2A/PCINT3)
16	PB2 (SS/OC1B/PCINT2)
15	PB1 (OC1A/PCINT1)

Arduino function

analog input 5
analog input 4
analog input 3
analog input 2
analog input 1
analog input 0
GND
analog reference
VCC
digital pin 13
digital pin 12
digital pin 11(PWM)
digital pin 10 (PWM)
digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MISO, MOSI, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Dodane biblioteke

Da možemo programirati sve senzore, potrebne su nam dodatne biblioteke:

Adafruit_MPL115A2-master

https://github.com/adafruit/Adafruit_MPL115A2

DHT

<https://github.com/adafruit/DHT-sensor-library>

U Arduino IDE-u pod Sketch odabrati Import Library-Add Library, i odabrati željenu datoteku. Restartati IDE i paziti da se glavni folder zove kao i fajlovi unutra.

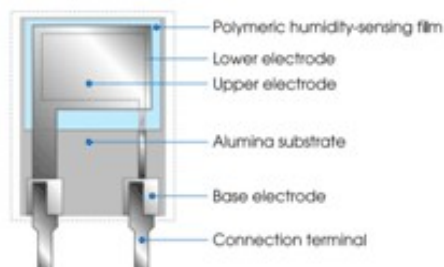
Elektronski blok – senzor vlažnosti i temperature [Nabava]

Senzor DHT11 mjeri temperaturu i vlažnost, temperaturu u rasponu 0-50 °C +- 2°C, a vlažnost u rasponu 20-95% RVZ s greškom od 5%. RVZ - *Relativna vlaga zraka je broj koji pokazuje odnos između količine vodene pare koja stvarno postoji u zraku u nekom trenutku i maksimalne količine vodene pare koju bi taj zrak na toj temperaturi mogao primiti da bi bio zasićen* [Izvor]. Napon pod kojim radi je 3.3 do 5V, što se na Uno ploči može mijenjati dostupnim prekidačem (sklopkom). Interval uzorkovanja je jedna sekunda odnosno Hz.

Ovaj senzor funkcionira po principu efekta električnog kapaciteta pa se tako naziva kapacitivnim senzorom [Više].

Kondenzator kapaciteta grade dvije metalne ploče između kojih se nalazi dielektrični materijal. Ukoliko se na površinu ploče ili udaljenost između ploča utječe nekom neelektričnom veličinom tada kapacitet C ovisi o toj veličini čime se dobiva kapacitivni senzor [preporuka za daljnje čitanje: Više]. Kod ovih senzora između elektroda se nalazi higroskopni materijal poput plastike ili polimera s tipičnom dielektričnom konstantom između 2 i 15. Pri normalnoj sobnoj temperaturi dielektrična konstanta vodene pare ima vrijednost oko 80 što je puno više od konstante

dielektričnog materijala senzora pa tako dolazi do povećanja kapaciteta senzora. Također, količina te vlage ovisi i o temperaturi okoline kao i o tlaku vodene pare. Na slici je prikazana temeljna struktura ovakvog senzora [[Izvor](#)]:

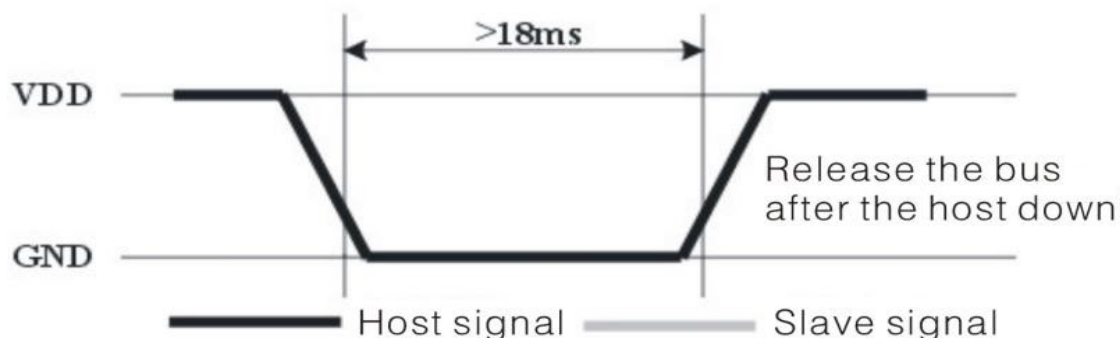


Niža elektroda je napravljena od zlata, platine ili drugog materijala a obavijena je polimernim slojem koji je osjetljiv na vlagu. Na tom polimernom filmu se nalazi i sloj zlata koji se ponaša kao gornja elektroda, koja isto dozvoljava vlazi prolazak do osjetilnog sloja. Komunikacijski proces i informacije dostupni su npr. na ovoj stranici [Wikitead](#).

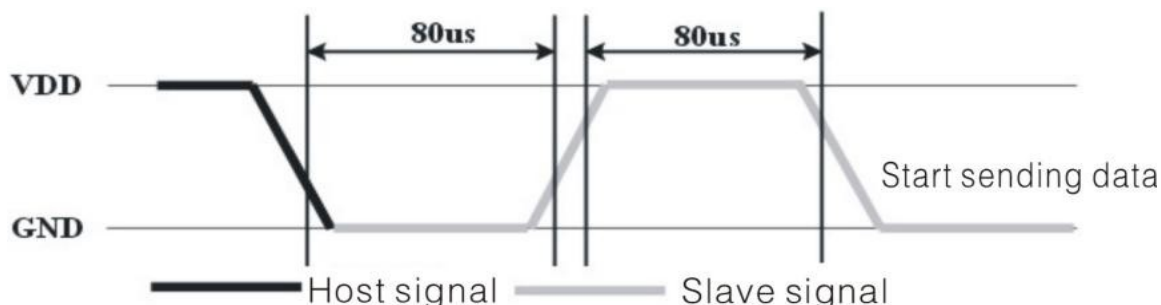
Proces komunikacije u koracima [izvor slika i preporuka radi izvrsno sumiranih podataka o senzoru [Pdf](#)]:

1. Nakon pokretanja senzora testira se temperatura i vlažnost okruženja, podaci se bilježe pri čemu podatkovne linije moraju biti održavane u stanju jedinice; DHT11 DATA pin je sada u input stanju – trenutku detekcije vanjskih signala.

2. MCU I/O postavljen je na output i u nisko-naponskom stanju, što može trajati najmanje 18ms nakon čega prelazi na input (to radi pull-up otpornik) i kao i podatkovne linije DHT11 u visoko-naponsko stanje, čekajući da DHT11 odgovori na signal, šaljući signal kao na slici:



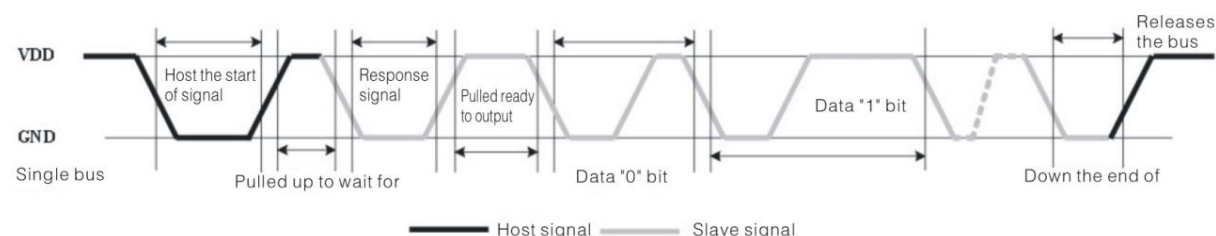
3. DATA pin je u nisko-naponskom stanju jer se čeka kraj zadržske DHT11 DATA pina u output stanju kako bi DHT11 mogao sigurno detektirati početni signal. Kada ga primi od MCU čekaći će po 80 mikrosekundi kako bi slao odgovor dok ne stane početni signal. Kada on stane počinje se čitati odgovor od DHT11.



4. Pri slanju podataka svaki bit informacije počinje na 50 mikrosekundnom nisko-naponskom stanju a duljina narednog signala visoko-naponskog stanja (za čiju promjenu se dodaje 26-28 mikrosekundi) određuje hoće li podatak biti '0' ili '1'. Nakon posljednjeg poslanog bita informacije DHT11 će čekati 50-70 mikrosekundi, a zatim će pull-up resistor osloboditi sabirnicu bus u free status.



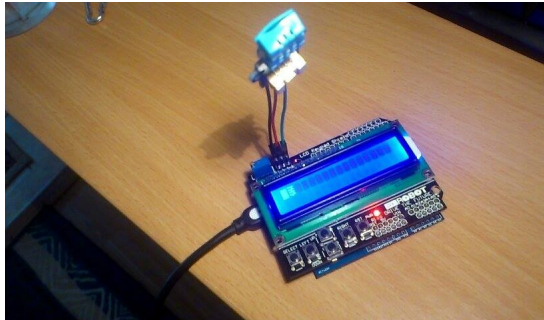
Slijedi cijeli dijagram:



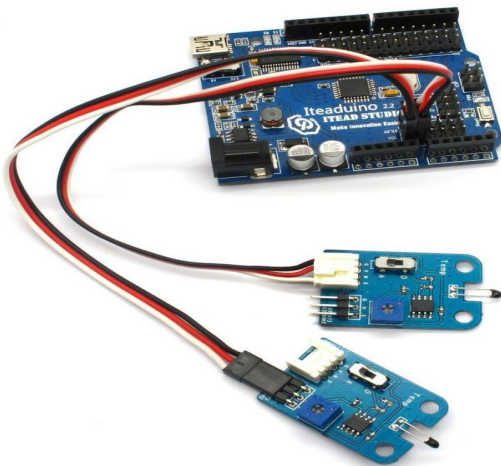
Ovaj senzor se može primijeniti kod grijanja, ventilacije, klimatizacije, uređaja za regulaciju vlage, opreme za testiranje i inspekciju elemenata takvih primjena, u raznim komercijalnim

proizvodima, meteo-stanicama, u medicinskoj opremi odnosno svugdje gdje je potrebno mjerenje/kontrola temperature i vlage.

Primjer potpuno pogrešnog spajanja (DHT11 na ICSP header) koje je uzrokovalo disfunkciju senzora (da nekome ne bi palo na pamet :)):

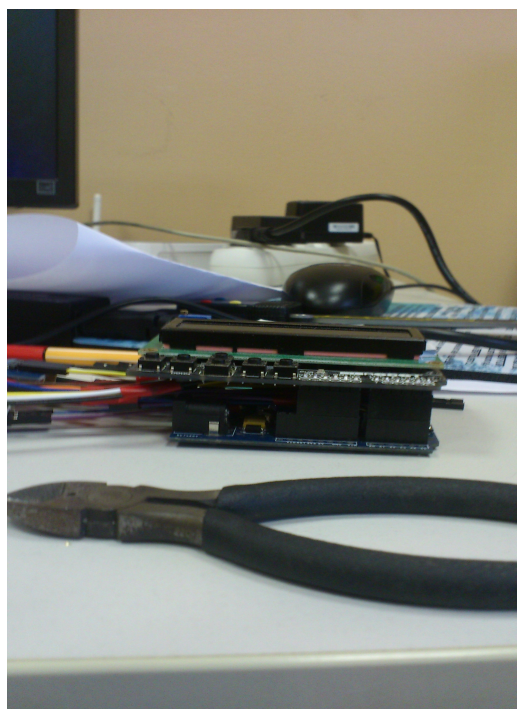
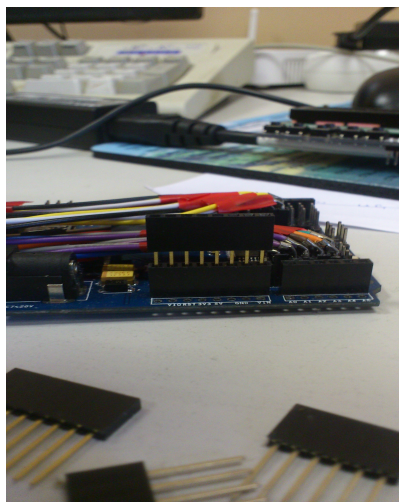


O temi spajanja DHT11 može se, kao i za ostale senzore pročitati klikom na wiki link koji se nalazi na dnu web stranice Mikrotrona pojedinog elementa ([mikrotron](#)). Wiki na slici prikazuje kako bi DHT11 trebao biti spojen na analogne pinove:



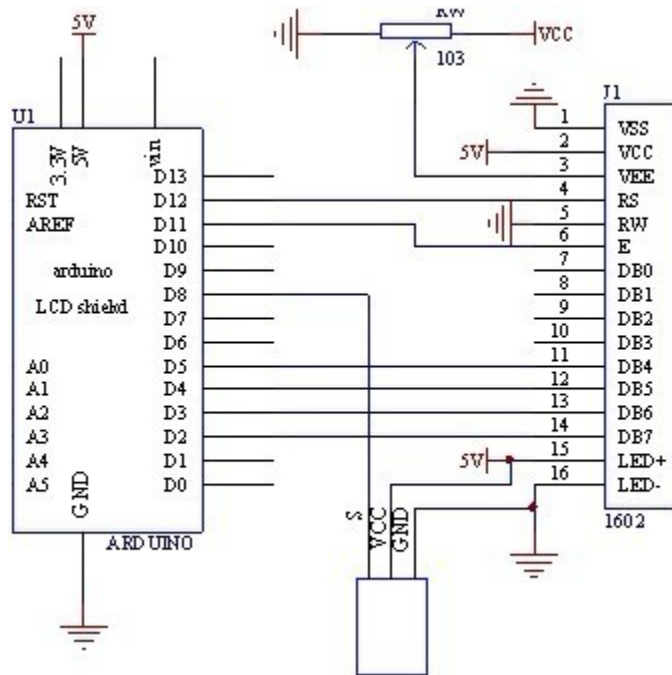
Nažalost, onda konektori smetaju montaži LCD shield-a.

Onda je potrebno dodati header set između Iteaduina i LCD Shielda kako bismo omogućili više mjesta u pogledu visine za konektore, a standardni header set je potrebno skratiti 4.45 mm:



Analogni pinovi nisu dali rezultate – ispostavilo se da se info uopće ne odnosi na ovu komponentu (pouka - čitati pozornije :)).

Ovu komponentu treba spajati na Arduino digitalne, ne analogne pinove sa [ArduinoWiki](#). U ovoj fazi već treba potražiti kod i shemu i naći konkretne informacije: [geeetech](#)



```
#include "DHT.h"
#include <LiquidCrystal.h>
#define DHTPIN 8
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  dht.begin();
}
void loop() {
  ...
}
```

No i ne bi trebalo očekivati izravno rješenje za sve 1/1 na jednoj stranici na internetu. Ta stranica kao i sve druge kada pokazuju sheme i kod navode da kada predajete parametar `DHTPIN` u `DHT dht(DHTPIN, DHTTYPE);` znači vrijednost pina na koji ste spojili DHT11, u ovom slučaju (`#define DHTPIN 8`) pin 8, vrijednost je ista kao broj pina. Ali s obzirom da digitalni pinovi počinju od nule, 5 je na primjer zapravo 4: DS4=5, DS5=6, DS6=7, DS7=8.

A sada se vratimo na primjer s vrijednošću 8. Isproban je ali je iznenada sve je pošlo po krivu – dobiju se sljedeće greške (slijedi samo nekoliko njih):

```
avrdude: stk500_getsync(): not in sync: resp=0x00
```

```
avrdude: stk500_getsync() attempt 10 of 10: not in sync: resp=0x03
```

```
avrdude: stk500_cmd(): programmer is out of sync
```

```
avrdude: stk500_initialize(): (a) protocol error, expect=0x14,  
resp=0x17
```

i svaki programer na internetu kaže isto – iznenada. Popravci su užasni, i iskreno glede tog „iznenada“ – ako je nešto radilo dosad na nekom nivou – učitalo sketcheve, spajalo se ok, onda mora biti nešto što je učinjeno nedavno – dali je u pitanju možda opet krivo spajanje? Pa tako sve treba vratiti npr. na analogne pinove da se utvrdi hoće li konekcija biti opet ok i hoće li se kod učitavati skica. Skica se učitala. Ali moraju se koristiti digitalni pinovi, pa što se onda desilo? Kao što je ranije spomenuto, `LiquidCrystal(rs, enable, d4, d5, d6, d7)` u ovoj konfiguraciji je `LiquidCrystal lcd(8, 9, 4, 5, 6, 7)`; pa kada bi senzor bio spojen na pin 8 koji je zauzet za RS (Register Select) dobiju se te ružne greške a rješavanje istih po net rješenjima uključuje nepotrebne korake poput flashanja Arduinovog Bootloadera, i tako dalje... A kada se spoji na neki drugi digitalni pin – 5 – konačno sve proradi.



Glede koda često se koristi isti segment oko potrebnih definicija i uključivanja libraryja iz kojih su boldane funkcije koje su tu korištene:

```
#include "DHT.h"
#define DHTPIN 4
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
...
void setup() {
    Serial.begin(9600);
    ...
    dht.begin();
}

void loop() {

    int h = dht.readHumidity();
    int t = dht.readTemperature(); }
```

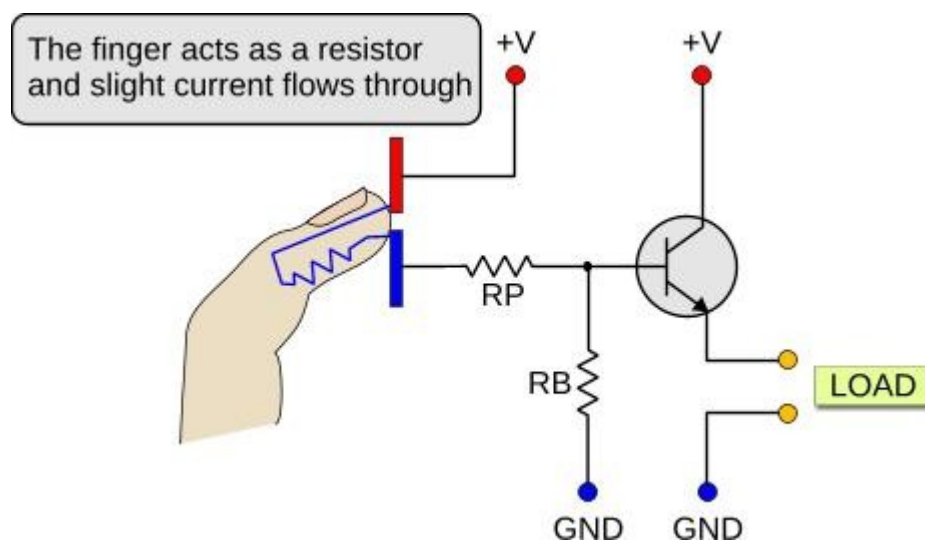

Elektronski blok – senzor dodira/prekidač [\[Nabava\]](#)

Kada nema dodira digitalni output je VDD a kada ima onda je 0V, tako da se prikaz rezultata aktivira na LOW, a obrnuto na HIGH. Napon pod kojim radi je 3.3 do 5V. [\[Pdf\]](#)

Senzor dodira može biti *otporni*, *kapacitivni*, površinski valovi itd.

Kod otpornog kada korisnik dotakne površinu bilježi se promjena u struji koja teče kroz zaslon. Isti se sastoji od nekoliko slojeva ali su najbitnija dva tanka električki vodljiva sloja, odvojena rasporedom dovoljno uskim da se ti metalni slojevi pritiskom točke prstom na ploču spoje pa se ploča počne ponašati u tom trenutku kao djelatelj napona s priključenim izlazom. To pak uzrokuje promjene u električnoj struji čime se registrira događaj i obrađuje dalje.

Ili se može ostvariti tako da ljudski prst zbog količine vode i soli predstavlja dielektrik pa kada dotakne obje ploče kroz njega proteče nešto struje do npr. tranzistora kao na sljedećoj slici [\[pic\]](#):



Ukratko ova vrsta senzora na dodir može biti implementirana na razne načine.

Kapacitivni senzor ima ploču obloženu materijalom koja sadrži električne naboje pa tako krugovi koji se nalaze na svakom uglu ploče mjere naboj i šalju informaciju na obradu. S obzirom da je i ljudsko tijelo električki vodljivo dodir takvog zaslona narušiti će elektrostatsko polje zaslona, što će se mjeriti upravo kao promjena kapaciteta.

Primjena senzora na dodir je stvarno raznolika, od pokretanja aparata za kavu, odnosno svih kuhinjskih aparata, svih aparata općenito s obzirom na potrebu za imitacijom rada mehaničke sklopke odnosno digitalnog 0/1. Ovo je možda i najupotrebljiviji senzor na svijetu.

Kod je prilično jednostavan i primjenjiv za druge senzore i nisu korištene nikakve posebne funkcije da bi se ovdje zasebno izdvajao taj dio koda. Zapravo se samo funkcijom `digitalRead` kojoj se predaje kao parametar vrijednost odabranog pina ispituje 'aktivnost' odnosno 'neaktivnost' senzora:

```
if(digitalRead(touchPin)==HIGH)
```

```
if(digitalRead(touchPin)==LOW)
```

Ovisno o tome dali je dodir detektiran ovdje se to samo ispisuje na lcd no naravno može se napraviti dalje što se želi. Za više pogledati [DFR](#).

Elektronski blok – zvučni senzor i mikrofon [\[Nabava\]](#)

Napon pod kojim radi je 4.5 do 5.5V. Osjetljivost mu je -50 dB što je jednako 3.1623 mV/Pa [\[Conversion\]](#).



Mikrofoni (slika [pic](#)) su elektroakustički pretvarači koji pretvaraju promjenu zvučnog tlaka u električki signal, napon ili struju. Njihova osjetljivost nam govori koliku ćemo imati promjenu električkog signala za danu promjenu zvučnog tlaka, a mjeri se u mV/Pa na frekvenciji od 1 kHz. Npr., uz zvučni tlak od 80 dB, ako je osjetljivost -60 dB (V/Pa) na priključnicama mikrofona se može dobiti razina napona od 1 mV. Ova razina može biti i veća ako se upotrijebi mikrofonsko pretpojačalo. Pri -60 dB, osjetljivost mikrofona bi bila 1 mV/Pa. Frekvencijska karakteristika osjetljivosti mikrofona nam govori kako njegova osjetljivost ovisi o frekvenciji. Idealni mikrofon bi imao jednaku osjetljivost od 20 Hz do 20 kHz. [\[MIC1\]](#)

Čisto za primjer, za zvukove koje je 'teže' detektirati koriste se oni velike osjetljivosti npr. -35 dB, a da se normalnim tonom glasa govori u mikrofon mogao bi se koristiti neki manje osjetljivosti npr. -60 dB. Također ako mjerimo osjetljivost mikrofona udaljeni 1m od njega, i udaljimo se za još jedan dakle za duplo osjetljivost će padati po 6 dB.

Općenito, mikrofoni se mogu izvesti na razne načine: ugljeni, dinamički, kondenzatorski, kristalni [\[MIC2\]](#). Ovdje govorimo o mikrofону

srednje osjetljivosti [[MIC3](#)]. S obzirom na opširnost teme preporuča se samostalno daljnje istraživanje, evo za primjer zanimljiv izvor informacija [Više1](#) ,[Više2](#) .

Naš senzor smo spojili na analogni pin A1. Na iteadeaduovom datasheetu za primjer se spaja na A0 ali s obzirom da naš LCD s A0 'hvata' evente gumbiju onda je to bolje ne raditi.

Kako bismo utvrdili promjenu zvuka potrebno je odrediti veličinu očekivane minimalne okolišne buke (na serial monitoru vidimo kako se senzor ponaša uz pomoć funkcije analogRead(A1) koja ce nam trebati i za ostalo u kodu), u ulozi tišine, jer ćemo tu vrijednost uspoređivati onda s očitanim da bi se provjerilo da li je došlo do nekog zvuka.

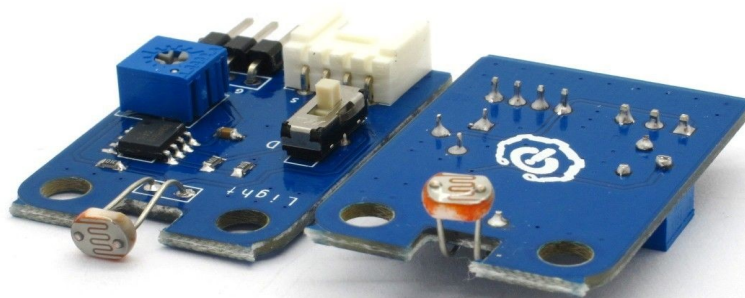
Kod ovog senzora, kada bi se tražila u komercijalnom izdanju detekcija zvuka, bi to bilo potrebno učiniti kod svakog klijenta zasebno, jer blaga ili snažna ili uopće promjena zvuka je relativan pojam zbog uvjeta okoline u kojoj se klijent nalazi, tu ulogu bi jednostavno odradio potencijometar. Veći okolni šum odnosno buku će prije detektirati onaj niske osjetljivosti nego visoke.

Kod svih analognih veličina koje se ovdje ispituju uvijek mora postojati okolišni minimum prema kojem se referira promjena.

Kod je kao i kod prethodnog senzora, glede same detekcije jednostavan jer se zapravo samo očitava stanje pina A1 i uspoređuje s 'tišinom' kako bi se dobila informacija o detekciji.

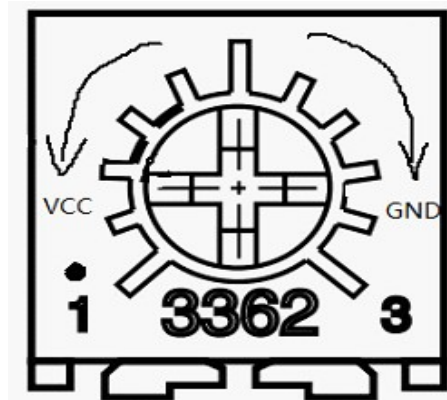
Elektronski blok – svjetlosni senzor [[Nabava](#)]

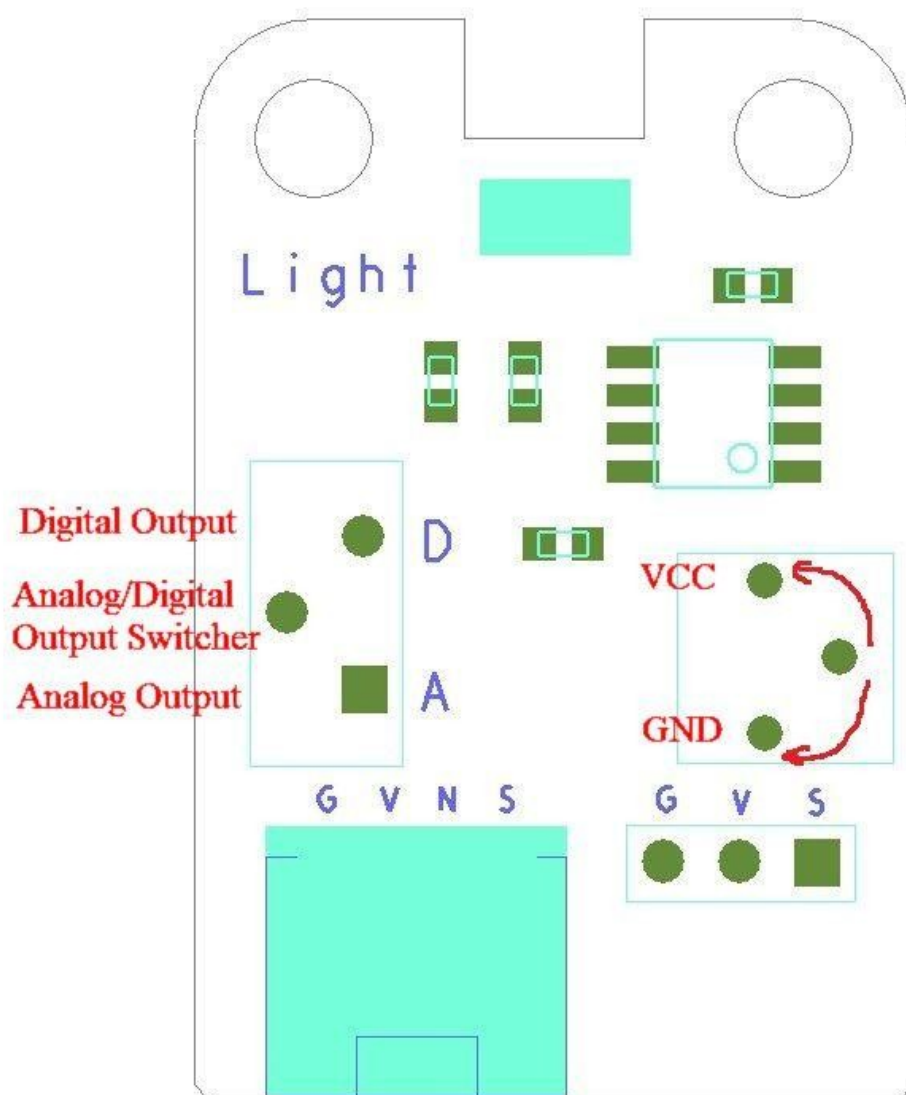
Nap
on pod



kojim je radi je 3.3 ili 5V. U našem projektu spojen je na A2. Ima, kao što se vidi na slici sklopku koja se može promijeniti s A (analogno) na D (digitalno). Ovisno o tome, ako je postavljena na analogno moći će se detektirati precizne informacije o detekciji svjetla u voltaži, a ako se stavi na digitalno moći će se očitati samo nagla promjena svjetla odnosno prisutnost ili nedostatak. Mi smo stavili na analogno ali s obzirom da u kodu samo ispitujem da li se promijenio nivo svjetlosti možda se moglo za takvu primjenu staviti i na digitalno. Dakle za sve konkretnije primjene, blage promjene definitivno treba biti na analognom.

Također sensor (fotootpornik) se može pomicati odnosno usmjeravati za bolju detekciju. Postoji i potencijometar kojime se povećava početno stanje odnosno (eng.threshold) iznos prema kojem se definira promjena stanja. Okretom udesno se povećava a u lijevo smanjuje.





Fotootpornik na sljedećoj slici [Pic] objašnjava njegov rad – njegov el.otpor smanjuje se s povećanjem intenziteta ulazne svjetlosti.

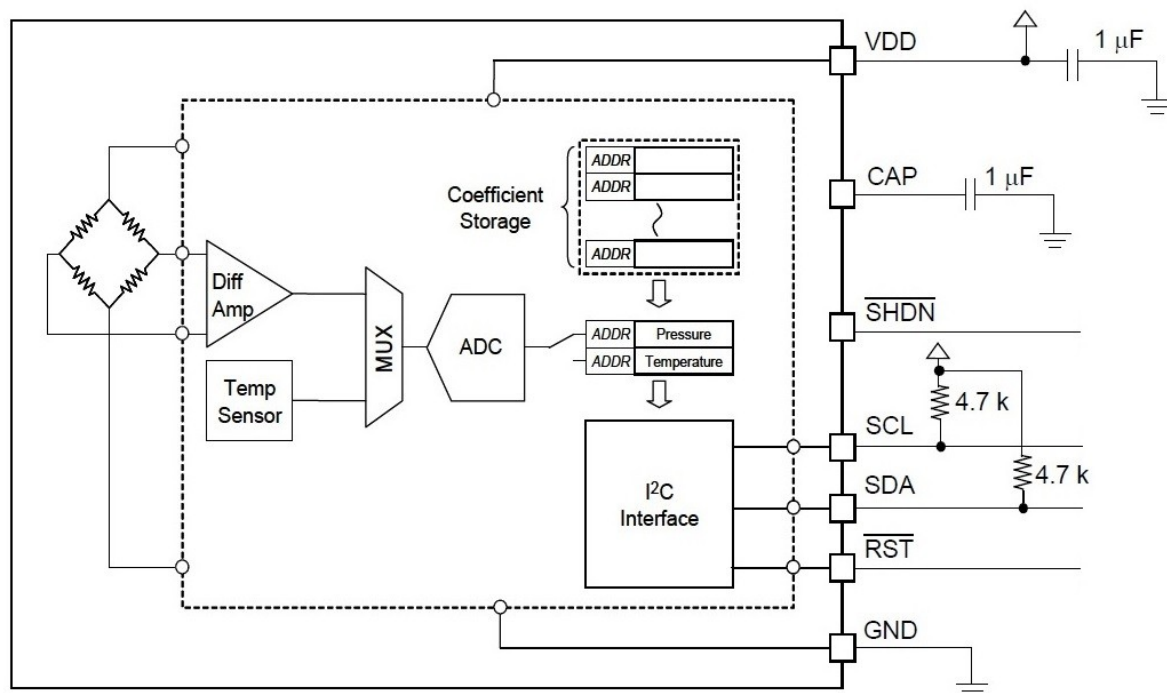


MPL115A2 – I2C senzor atmosferskog pritiska/temperature

[[Nabava](#)]

Radni napon je 2.4 do 5.5V. Mjerni raspon: 500-1150 hPa (do 10 km visine). Rezolucija: 1.5 hPa/50 m. Koristi I2C 7-bitnu adresu 0x60.

MPL115A2 Breakout Board



Gornja slika dolazi iz pdf-a kojeg ne bi bilo loše pogledati [[PDF](#)].

Glede mjerenja temperature se baš i ne preporuča pa u našem primjeru na LCD-u se rezultat tog mjerenja ne prikazuje.

Spojeni su na A5(svi osim SDA) i A4(samo SDA)[[Spoj](#)]. Ovdje je uključen library Adafruita kako bi se pojednostavio proces programiranja. Cijelo vrijeme s raznim dostupnim testnim kodovima, koji se inače programiraju svi na isti način (jedino su neki 'uljepšani' uključivanjem libraryja pa time korištenjem lijepih funkcija) senzor nije ništa očitavao, točnije rečeno pokazivao je vrijednosti koje su predefinirane (izračunom formule koja će uskoro biti prikazana), ali indikativnije, ako se promjenom mehaničkog postupka sa senzorom ništa ne mijenja na serial monitoru

znači da detekcije nema. Sve dostupne testne kodove provjerili smo liniju po liniju pokušavajući naći grešku, i greške nije bilo niti zapravo može biti jer se s ovim senzorom to radi po unaprijed zadanim pravilima, dobro definiranim već na samom datasheetu. Slučajno prihvatom senzora u ruku, točnije indirektnim pritiskom na spoj s jumpericama primjećeno je da dolazi do (očekivanih) vrijednosti i da se podaci mijenjaju kako mičemo sklop. Nakon izvjesnog izgubljenog vremena svelo se na to da se gubi kontakt na tom mjestu pa se to pričvrstilo i problem je nestao.

Cijela logika računanja se ukratko, s obzirom da je iscrpno objašnjeno na samom datasheetu, svodi na izračun formule:

$$\text{Pressure (kPa)} = P_{\text{comp}} * \left[\frac{115 - 50}{1023} \right] + 50$$

10-bitni kompenzirana visina tlaka se računa po:

$$P_{\text{comp}} = a_0 + (b_1 + c_{12} * T_{\text{adc}}) * P_{\text{adc}} + b_2 * T_{\text{adc}}$$

P_{adc} je 10-bitna ADC visina tlaka MPL115A -a

T_{adc} je 10-bitna ADC visina temperature MPL115A -a

a_0 je koeficijent tlačnog odmaka

b_1 je koeficijent tlačne osjetljivosti

b_2 je koeficijent temperaturnog odmaka

c_{12} je koeficijent temperaturne osjetljivosti

Tijek je općenito sljedeći:

1. Čitanje koeficijenata koji se mijenja od ploče do ploče, a spremljeni su obično u lokalnoj memoriji mikrokontrolera, ali kada se dobiju na toj ploči se više ne mijenjaju pa ga u domicilni mikrokontroler ne treba spremati više puta, a u modu čitanja slave uređaj nije aktivan.

2. Pretvorba podataka čiji se rezultat sprema u registre, uzimaju se očitavanja sa MPL115A i opet pretvaraju podaci...

Pcomp će imati vrijednost 0 s ulaznim tlakom od 50 kPa a 1023 (maksimalnu vrijednost) s ulaznim tlakom od 115 kPa.

Prije svega treba uključiti u našem slučaju:

```
#include <Adafruit_MPL115A2.h>
```

```
#include <Wire.h>
```

stvoriti objekt klase čije ćemo funkcije koristiti:

```
Adafruit_MPL115A2 mpl115a2;
```

i preostaje praktički samo poziv funkcije `mp115a2.getPressure()`;

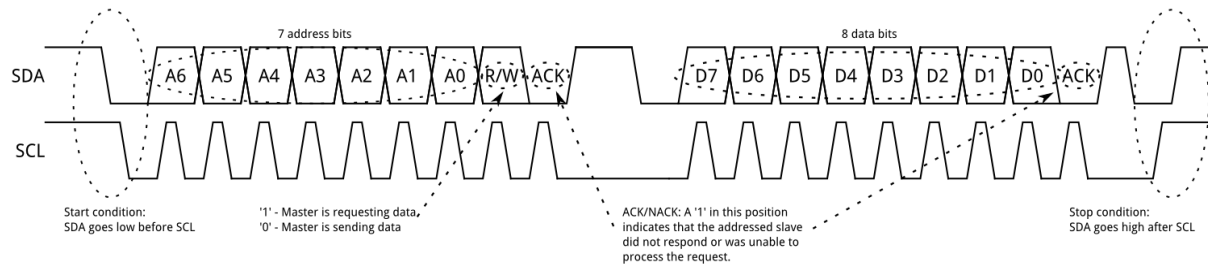
koja vraća float vrijednost tlaka.

S obzirom da se ovdje radi o I2C komunikaciji koristi se niz funkcija iz Wire.h koje je nužno koristiti kako bi se ona ostvarila. Također kako bi se komunikacija pojednostavila dogovorene su I2C naredbe:

Device Address + write bit "To Write"	0xC0
---------------------------------------	------

Device Address + read bit "To Read"	0xC1
Command to Write "Convert Pressure and Temperature"	0x12*
Command to Read "Pressure ADC High byte"	0x00
Command to Read "Pressure ADC Low byte"	0x01
Command to Read "Temperature ADC High byte"	0x02
Command to Read "Temperature ADC Low byte"	0x03
Command to Read "Coefficient data byte a0_MSB"	0x04
Command to Read "Coefficient data byte a0_LSB"	0x05
Command to Read "Coefficient data byte b1_MSB"	0x06
Command to Read "Coefficient data byte b1_LSB"	0x07
Command to Read "Coefficient data byte b2_MSB"	0x08
Command to Read "Coefficient data byte b2_LSB"	0x09
Command to Read "Coefficient data byte c12_MSB"	0x0A
Command to Read "Coefficient data byte c12_LSB"	0x0B

Glede I2C na hardverskom nivou svaka I2C sabirnica ima dva signala: SCL i SDA. SCL je clock signal, SDA data signal. Sam protokol je kompleksan ali u osnovi poruke se dijele na dva dijela, jedan adresni i jedan ili više podatkovnih. Na adresnom master indicira slavea kojemu se poruka šalje, a na podatkovnom dijelu se 8-bitne poruke prenose s mastera na slavea ili obrnuto. Podatak se stavlja na SDA liniju kada se SCL spusti, a uzorkuje se kada se SCL digne, a slika ispod to lijepo prikazuje [[Izvor](#)]. Vrijeme za sve to ovisi o uređaju.



Primjer upotrebe funkcija iz Wire.h:

```
...

Wire.beginTransmission(BarometerAddress); //CONVERSION START
Wire.write(byte(0x12));
Wire.write(byte(0x00));
Wire.endTransmission();

...
```

Cijeli kod u takvom obliku možete vidjeti i isprobati na ovoj stranici:
[KitronikCode](#).

Ako se ne prikazuje očitavanje tlaka problem može biti u kontaktu između senzora i konektora pa pričvrstite žicom kao i mi.

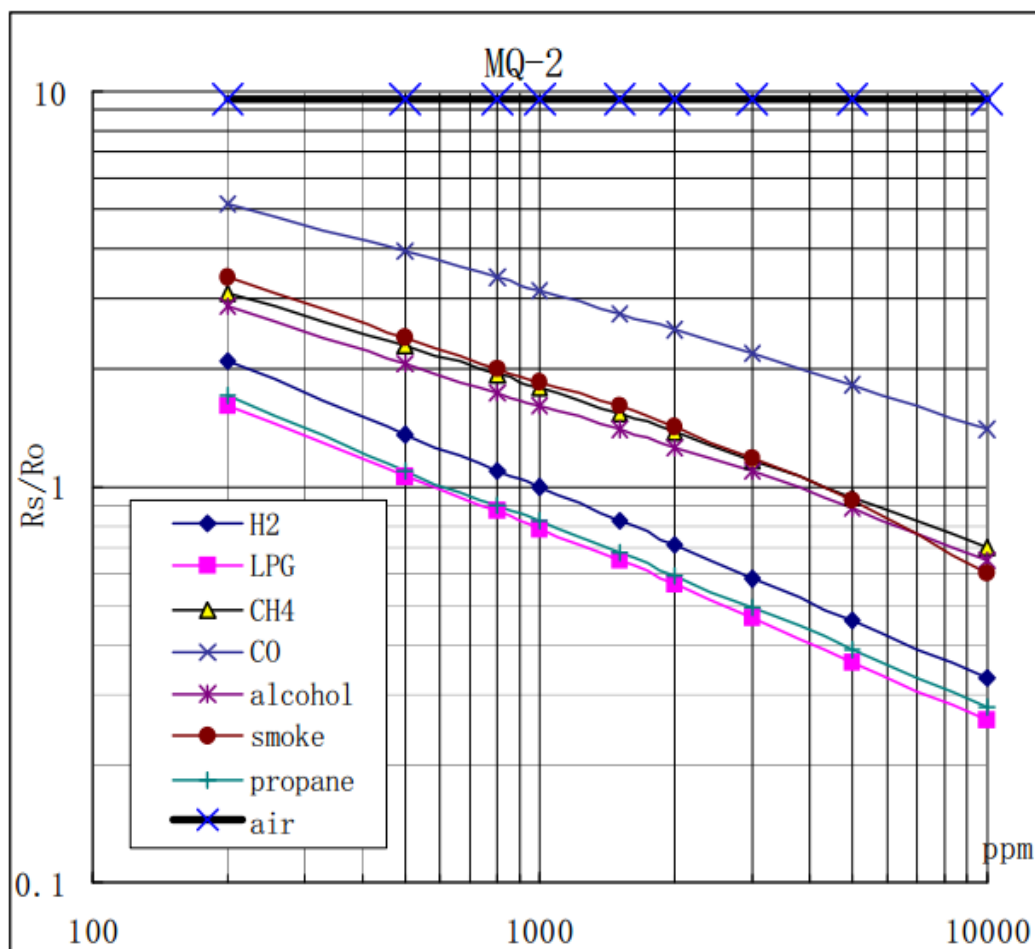
Elektronski blok – MQ-2 senzor plinova [\[Nabava\]](#)

Radni napon je 5V. Analogni je senzor. Detektira vodik, izobutan, ukapljeni naftni plin, metan, ugljični monoksid, alkohol, dim, propan i druge štetne plinove [\[WIKI\]](#).



Ovdje testni kod ne definira koji je plin detektiran već samo da je detektiran, što znači da ovdje ima mjesta za proširenje koda u priličnom obimu. Spojen je na pin A3.

Dijagram je preuzet iz službenog datasheeta i zapravo govori mnogo, s obzirom da je nakon spajanja testiran na način da je pušten kraj senzora plin s upaljača, i detektiran, a onda otvorena pokraj bočica sa 70%-tnim alkoholom što nije detektirao. Zanimljivo, reagirao je na otvorenu bocu rakije.



Kod je dakle pisan samo kao svaki testni da utvrdi prema oglednoj fiksnoj vrijednosti promjenu, kao i kod prethodnih senzora.



Gornji sloj je načinjen od nehrđajuće čelične mrežice koja onemogućava ikakvim česticama osim elementima plina da prođu do senzora, također štiti senzor, pri visokim temperaturama i tlakovima plinova neće doći do oštećenje modula senzora, dva od 6 vodiča se koriste za prijenos topline do osjetilnog elementa, a ostali za dohvaćanje signala. Kod tog osjetilnog elementa se, djelovanjem struje koja preko vodiča stvara toplinu, plinovi koji do njega dođu ioniziraju te ih element apsorbira. To mijenja otpor osjetilnog elementa što pak mijenja vrijednost struje koja iz njega izlazi [[Izvor](#)].

PIR senzorski modul [[Nabava](#)]

Ovdje smo naš PIR spojili na digitalni pin D4. Logička jedinica nastaje kada se detektira pokret. Kod je, opet, vrlo jednostavan s obzirom da se samo detektira pokret, znači opet

...

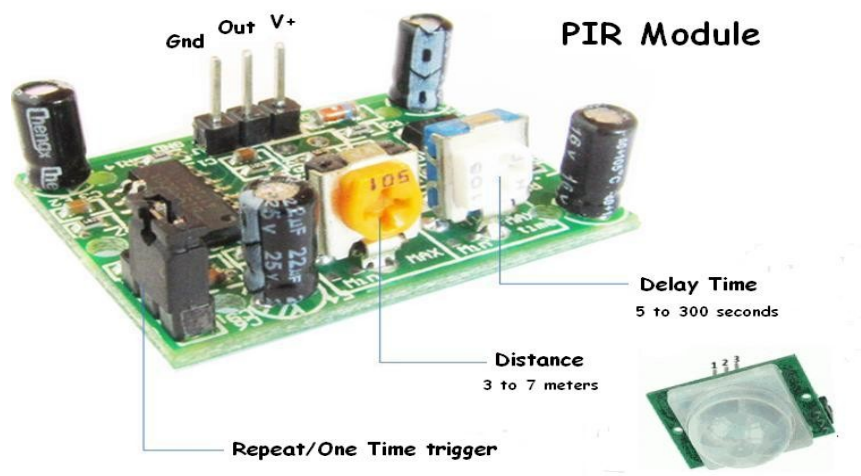
```
if(digitalRead(touchPin)==HIGH)
```

...

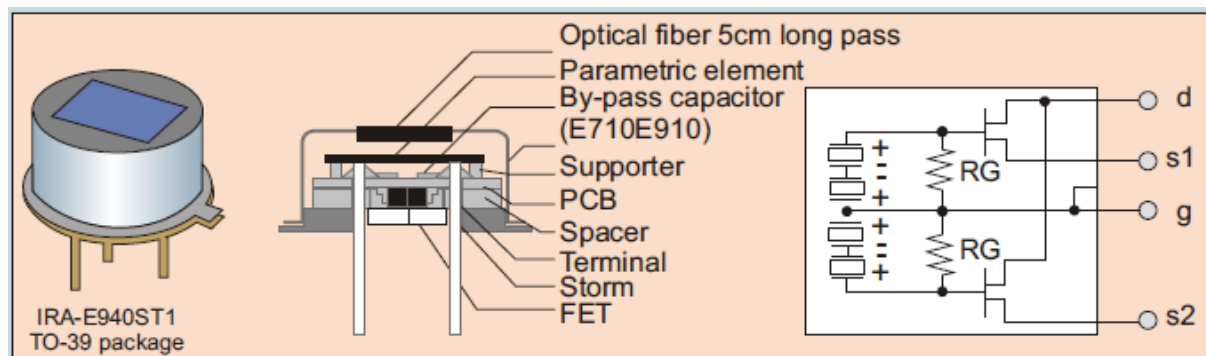
Radi jednostavnosti testiranja svoj smo PIR ograničili na 3m preko potencijometra. Sljedeća slika detaljno pokazuje što se sve može podesiti na ovom modulu. Ako se želi smanjiti udaljenost potencijometar treba okrenuti nalijevo, i obrnuto (min 3m, max 7m). Drugi potencijometar se odnosi na kašnjenje odnosno nakon koliko će se aktivirati, lijevo smanjivanje desno povećavanje (min 0.5 s, max 300 s).

L – okida se ali se ne ponavlja

H – okida se i ponavlja



Sva topla tijela (toplija od 273 C) emitiraju EM radijaciju u IR području. Pomoću sabirnog zrcala leće zrake se koncentriraju na piroelektrični pretvornik.



Detektira promjenu temperature, bez pokreta nema promjene temperature. Osjetljivi su na temp.kože osobe preko emitirane radijacije crnog tijela na srednje dugim IR valovima, u kontrastu na objekte u pozadini pri sobnoj temperaturi. Radijacija crnog tijela (Planckov zakon, toplinska radijacija) je vrsta EM radijacije unutar ili oko tijela u termodinamičkom ekvilibrijumu sa svojom okolinom, ili koja emitira crno tijelo (prozirno i nereflektivno tijelo) koje se drži na konstantnoj uniformnoj temperaturi. Toplinska radijacija koju spontano emitira mnogo običnih objekata može biti aproksimirana kao radijacija crnog tijela. Radijacija predstavlja pretvorbu tjelesne toplinske energije u EM energiju. Objekt koji apsorbira svu radijaciju kojom je obavljen na svim dužinama valova naziva se crno tijelo.

Sa senzora se ne emitira energija zato se zove pasivni. To ga razlikuje od električnog oka koji detektira doticaj s IR zrakama.

Koristi se kao intrusion senzor, za kontrolu svjetla, mjerenje temperature, detektor vatre, automatsko otvaranje vrata itd.

Idealan pokret bi trebao biti poprečan ne prema njemu jer PIR detektira promjene topline između zona; što se više zona prođe lakše mu je detektirati.

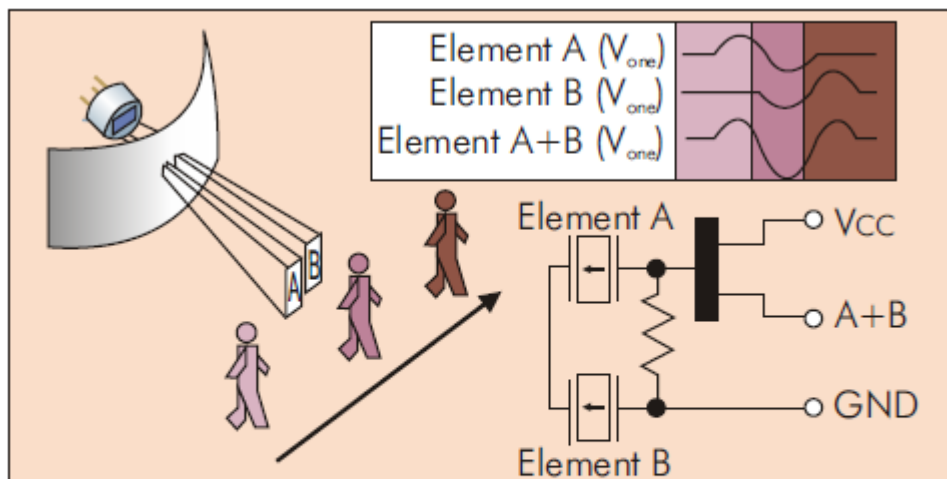


Fig. 2 : A PIR sensor works by sensing the difference in infra-red radiation between one sensing element and another

PIR senzor stvara privremeni ele.potencijal kada god se pojavi promjena u IR radijaciji. Osnovna struktura pasivne infracrvene naprave su Fresnel leće, PIR senzor, pojačalo, i komparator plus time delay struktura.

Fresnel leće fokusiraju IR radijaciju na PIR senzor pa PIR senzor mjeri promjenu u IR odnosu i stvara razliku ele.potencijala odgovarajuću varijaciji u IR radijaciji.

Piroelektričnost je sposobnost određenih materijala da generiraju privremeni ele.potencijal gdje su grijani ili hlađeni; ovaj efekt je promatran na kvartznim kristalima – ako kristal razvije pozitivan naboj s jedne strane tijekom grijanja razviti će negativan naboj na istoj strani tijekom hlađenja – ova sposobnost kristala koristi se za detekciju IR radijacije.

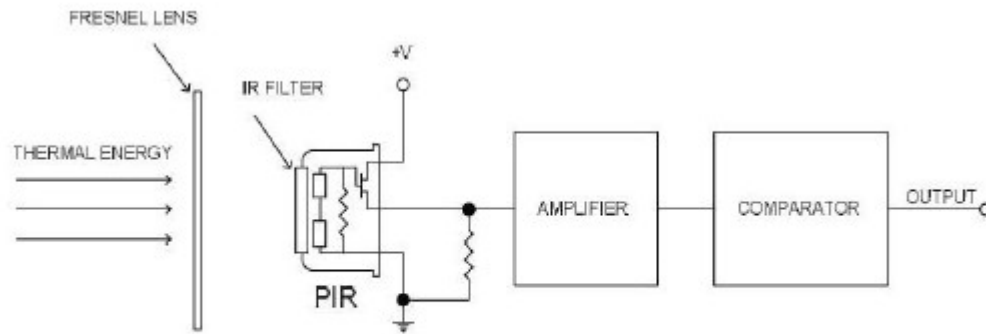


Figure 2.1: A typical PIR sensor block schema.

U zadnjoj fazi na komparatoru komparator daje 5V za logičku jedinicu a 0V za logičku nulu. Naboj na kristalima se pretvara u razliku el.potencijala preko FET tranzistora ugrađenog u senzor.