

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs
CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

Arduino – comunicatia I2C intre 2 placi

Ce reprezinta I2C ?

I2C este ,in explicatia lui, cea mai simpla un protocol prin care poti realiza transfer de date intre mai multe dispozitive. Spre exemplu, doresti ca in proiectul tau sa transmitsi la distanta, prin intermediul a 2 fire, pachete de date. Aceste pachete de date le vei transmite bit cu bit, unul dupa altul, catre unul sau mai multe dispozitive. Sau invers, poti receptiona si poti stoca pachete de date de la unul sau mai multe dispozitive.

Cum se conecteaza dispozitivele ?

In primul rand, ai aflat ca protocolul I2C iti permite sa transmitsi si sa receptionezi pachete de date sau bytes. In al doilea rand, le poti conecta sub forma unei retele pe o singura magistrala. Poti conecta in jur de 100 de placi Arduino folosind doar 2 fire, unul de date si unul de clock.

Conceptul de master si slave ?

Protocolul I2C introduce ideea ca o singura placa trebuie sa fie master, iar restul placilor trebuie sa fie slave-uri. Master-ul este cel care initializeaza comunicatia si efectueaza transferul de date din si catre slave-uri. Mai simplu spus, master-ul poate fi privit ca un server, iar slave-ul poate fi privit ca un client.

Analogic vorbind, clientul intotdeauna raspunde cererilor serverului. Daca serverul doreste date, acesta ii „spune“ clientului identificat printr-o adresa ca doreste acest lucru. Clientul se conformeaza si indeplineste cererea. In termeni generali, asa se realizeaza o tranzactie I2C intre mai multe placi Arduino.

Daca doresti sa studiezi mai multe despre protocolul I2C, nivele logice, cum se transmit serial octetii si alte informatii utile, acceseaza link-urile de mai jos:

<http://www.i2c-bus.org/>

http://www.robot-electronics.co.uk/acatalog/I2C_Tutorial.html

<https://learn.sparkfun.com/tutorials/i2c>

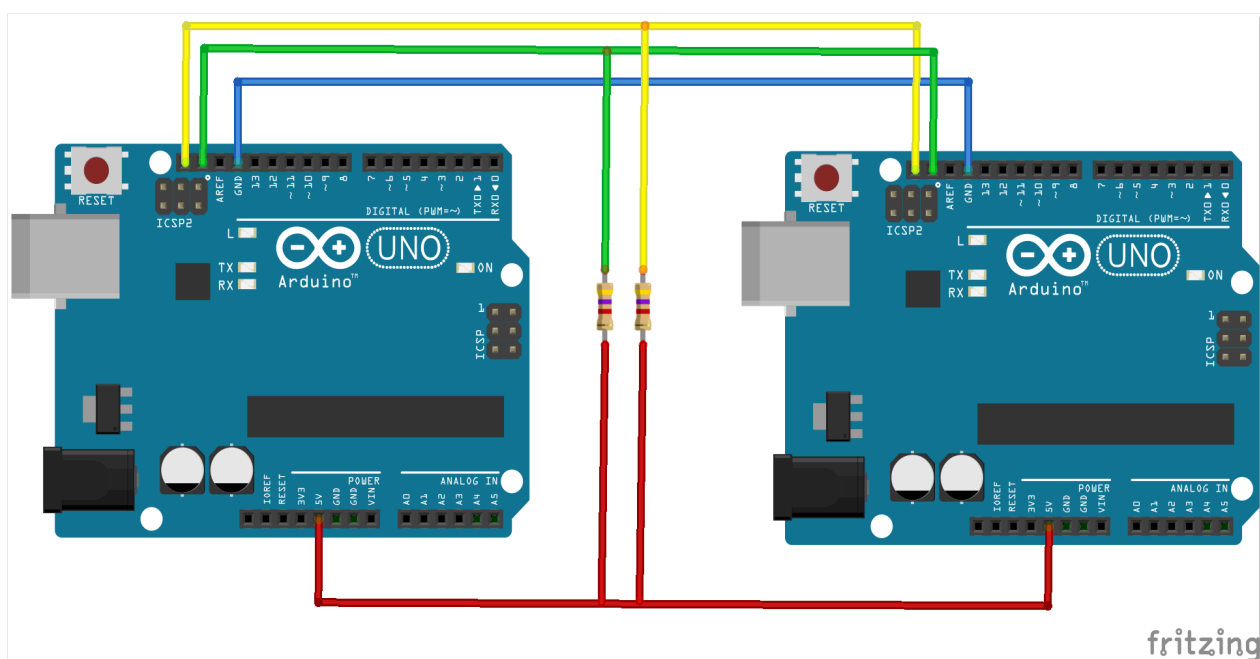
Cum conectez 2 placi Arduino ?

Pentru a conecta cele 2 placi Arduino vei avea nevoie de urmatoarele componente:

<http://www.robofun.ro/forum>

- 2 placi Arduino: <http://www.robofun.ro/arduino>
- Fire de conexiune: <http://www.robofun.ro/cabluri>
- Un breadboard: <http://www.robofun.ro/breadboard>
- Rezistoare: <http://www.robofun.ro/electronice/rezistoare>

Conecteaza placile Arduino folosindu-te de diagrama de mai jos:



În primul rând, trebuie să realizezi masă comună între cele 2 plăci, altfel nimic nu va funcționa corect. Pe diagrama masă comună este realizată prin firul de culoare albastru. Firul se afla conectat între pinii GND ale plăcilor Arduino.

De asemenea conectează pinii 5V folosind firul de culoare roșie. Lucrul asta este necesar pentru că trebuie să alimentezi rezistoarele de 4.7K.

Magistrala I2C este cea formată din cele 2 fire de culoare galbenă și albastru. Prin intermediul firelor vor circula biții și semnalele de tact.

Cum programez placile ?

În primul rând, trebuie să alegi care placă este master-ul și care placă este slave. Nu contează ce

placa alegi, cat timp cealalta va fi diferita. Spre exemplu, poti alege placa din stanga ca fiind placa master, iar placa din dreapta ca fiind placa slave.

Mai jos sunt listate 2 sketch-uri, unul il vei incarca in placa master, iar celalalt il vei incarca in placa slave.

Codul pentru placa master:

```
#include <Wire.h>

#define LED_PIN 13
byte x = 0;

void setup()
{
    Wire.begin();
    pinMode(LED_PIN, OUTPUT);
    digitalWrite(LED_PIN, LOW);
}

void loop()
{
    Wire.beginTransmission(9);
    Wire.send(x);
    Wire.endTransmission();
    x++;
    if (x > 5) x=0;
    delay(450);
}
```

Cum functioneaza sketch-ul placii master ?

Chiar daca la prima vedere protocolul I2C pare greoi si complicat, toate instructiunile din mediul Arduino sunt mult simplificate. Prima linie de cod este necesara, ea include biblioteca Wire, adica biblioteca responsabila comunicatiei I2C intre placi. In mediul Arduino se numeste Wire si nu I2C. Se declara LEDul aflat pe placa Arduino ca fiind conectat pe pinul 13 si se initializeaza o variabila de tip byte, variabila x.

In rutina setup() au loc mai multe procese si anume:

- se initializeaza placa master prin linia Wire.begin() fara niciun parametru intre paranteze (vei vedea ca la placa Slave este musai sa aplici si un parametru, adica adresa slave-ului).

- Se initializeaza pinul 13 ca fiind pin de OUTPUT si se stinge prin linia digitalWrite.

In rutina loop() au loc urmatoarele procese:

- Prin linia Wire.beginTransmission(9) se initializeaza comunicatia I2C cu placa slave. Parametrul 9 reprezinta adresa placii slave.
- Prin linia Wire.send(x) se transmite catre placa slave continutul variabilei x.
- Prin linia Wire.endTransmission() se incheie transferul I2C catre slave.
- Variabila x este incrementata, dupa care tot procesul explicat mai sus se reia, adica se va transmite catre placa slave continutul variabilei x, dar incrementat.

Codul pentru placa slave:

```
#include <Wire.h>

#define LED_PIN 13
#define LED_1 12
#define LED_2 11

int x;

void setup() {
  Wire.begin(9);
  Wire.onReceive(receiveEvent);

  pinMode(LED_PIN, OUTPUT);
  pinMode(LED_1, OUTPUT);
  pinMode(LED_2, OUTPUT);

  digitalWrite(LED_PIN, LOW);
  digitalWrite(LED_1, LOW);
  digitalWrite(LED_2, LOW);

  x = 0;
}
```

```

void loop() {

    if (x == 0) {
        digitalWrite(LED_1, HIGH);
        delay(200);
        digitalWrite(LED_1, LOW);
        delay(200);
    }

    if (x == 1) {
        digitalWrite(LED_2, HIGH);
        delay(200);
        digitalWrite(LED_2, LOW);
        delay(200);
    }
}

void receiveEvent(int howMany) {
    x = Wire.receive();
}

```

La nivelul plăcii Slave, lucrurile se petrec sub următoarea formă:

- Placa Arduino este inițializată ca și placa Slave cu adresa 9.
- La fiecare tranzacție I2C inițializată de către placa Master, placa Slave va executa rutina `receiveEvent(int howMany)`. Ea funcționează asemănător cu o întrerupere. Ori de câte ori Master-ul transmite „ceva“ se va executa această rutină, prin care se stochează în variabila `x` ceea ce a transmis placa Master.
- În rutina `loop()` placa Slave testează ceea ce a transmis placa Master și anume, dacă a transmis valoarea 0 atunci se va aprinde LEDul 1, iar dacă s-a transmis valoarea 2 atunci se va aprinde LEDul 2.

Acesta este un exemplu simplu, prin care o placă Arduino master transmite către o altă placă Arduino slave o serie de valori. Placa Slave reacționează diferit în funcție de valoarea transmisă.

Cum pot dezvolta acest proiect ?

In general protocolul I2C se utilizeaza la transmisia informatiei cu viteze cuprinse intre 10 si 400 Khz. Poti transmite orice, in functie de proiectul pe care doresti sa il abordezi.

Spre exemplu, poti conecta o memorie I2C direct la placa Arduino sau iti poti realiza un termometru digital care sa iti afiseze temperatura direct pe ecranul calculatorului:

<http://www.hobbytronics.co.uk/arduino-external-eprom>

<http://www.jeremyblum.com/2011/02/13/arduino-tutorial-7-i2c-and-processing/>

Nu uita ca exista o gama variata de senzori ce utilizeaza protocolul I2C pentru a comunica la viteze mari, senzori digitali de presiune, accelerometre, IMU, giroscopae, s.a.m.d.