

Universitatea Tehnică a Moldovei  
Facultatea Calculatoare Informatică și Microelectronică  
Departamentul Ingineria Software și Automatică

# RAPORT

Lucrarea de laborator nr. 7.1  
La disciplina “Internetul Lucrurilor”  
**Tema: I<sup>2</sup>C - Protocol HW**

A efectuat: st. gr. SI-211  
A verificat:

Adrian Chihai  
Valentina Astafi

**Chișinău – 2024**

## 1 Definirea problemei

Sa se realizeze o aplicatie ce va implementa comunicatiile intre echipamente dupa cum urmeaza:

1. Protocol fizic de comunicare - Comunicarea intre DOUA Microcontrollere prin interfata I<sup>2</sup>C

- MCU1 - implementeaza sensorul digital cu interfata I<sup>2</sup>C pentru sensorul ultrasonic HCS-04, unde se executa colectarea datelor de la interfata sensorului si se retransmite catre interfata I<sup>2</sup>C la detectarea unei cereri de citire a datelor.
- MCU2 - executa cererea prin interfata I<sup>2</sup>C catre sesorul digital ultrasonic (MCU+HCS-04) si afiseaza datele pe interata seriala

## 2 Descrierea funcțiilor programului

- **setup()**

**Inițializează comunicația serială și I2C.** Se configurează comunicația serială la o rată de 9600 bps și inițializează microcontrolerul ca master în comunicația I2C.

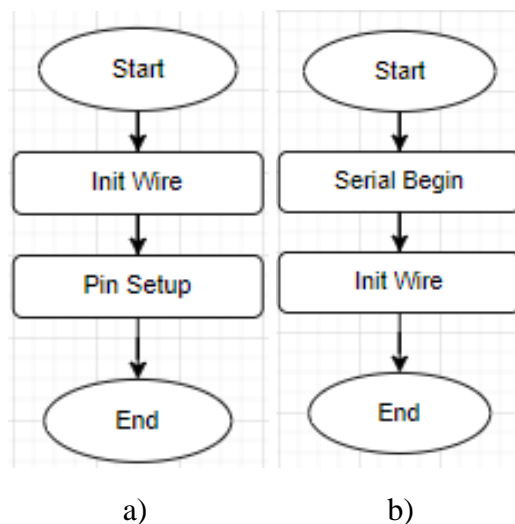
- **loop()**

**Solicită periodic date și așteaptă un răspuns.** Funcția apelează requestDistance() pentru a cere date de la microcontrolerul secundar, apoi așteaptă 500 de milisecunde până la următoarea solicitare.

- **requestDistance()**

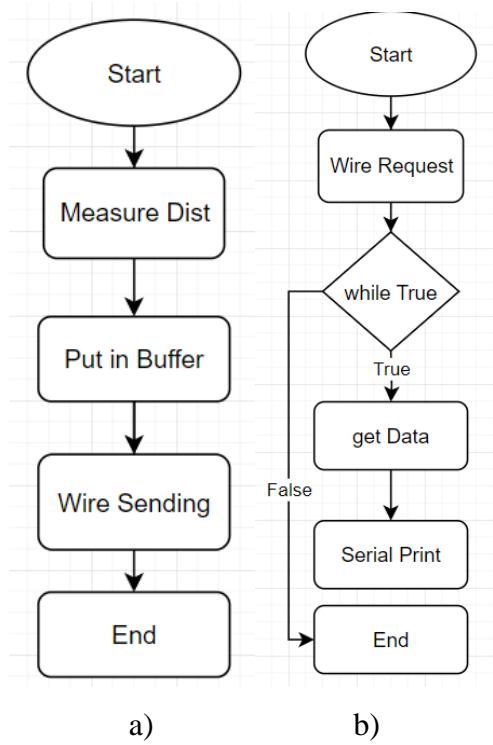
**Solicită și afișează distanța de la un senzor prin I2C.** Solicită un număr de 2 octeți de la microcontrolerul secundar, așteaptă până când toți octeții sunt disponibili, citește datele într-un buffer și afișează distanța calculată pe portul serial.

Figura 2.1 reprezintă funcția setup din program. După ce această funcție se termină, se sare la funcția de loop.



**Fig. 2.1.** Funcția *setup*: a pentru MCU1, b pentru MCU2

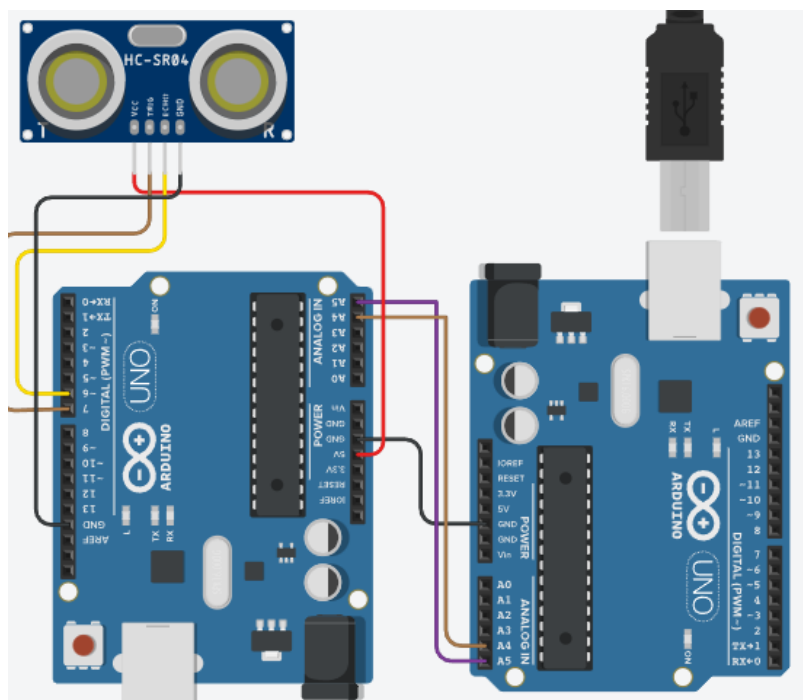
Figura 2.2 logica pentru funcțiile *request* și *send*.



**Fig. 2.2.** Funcțiile *send(a)*, *request(b)*

### 3.Circuitul elaborat

În figura 2.3. este realizat circuitul conform cerințelor



**Fig. 2.3.** Circuitul electric realizat

În figura 2.4. este circuitul pornit și putem observa cum senzorul capturează distanța

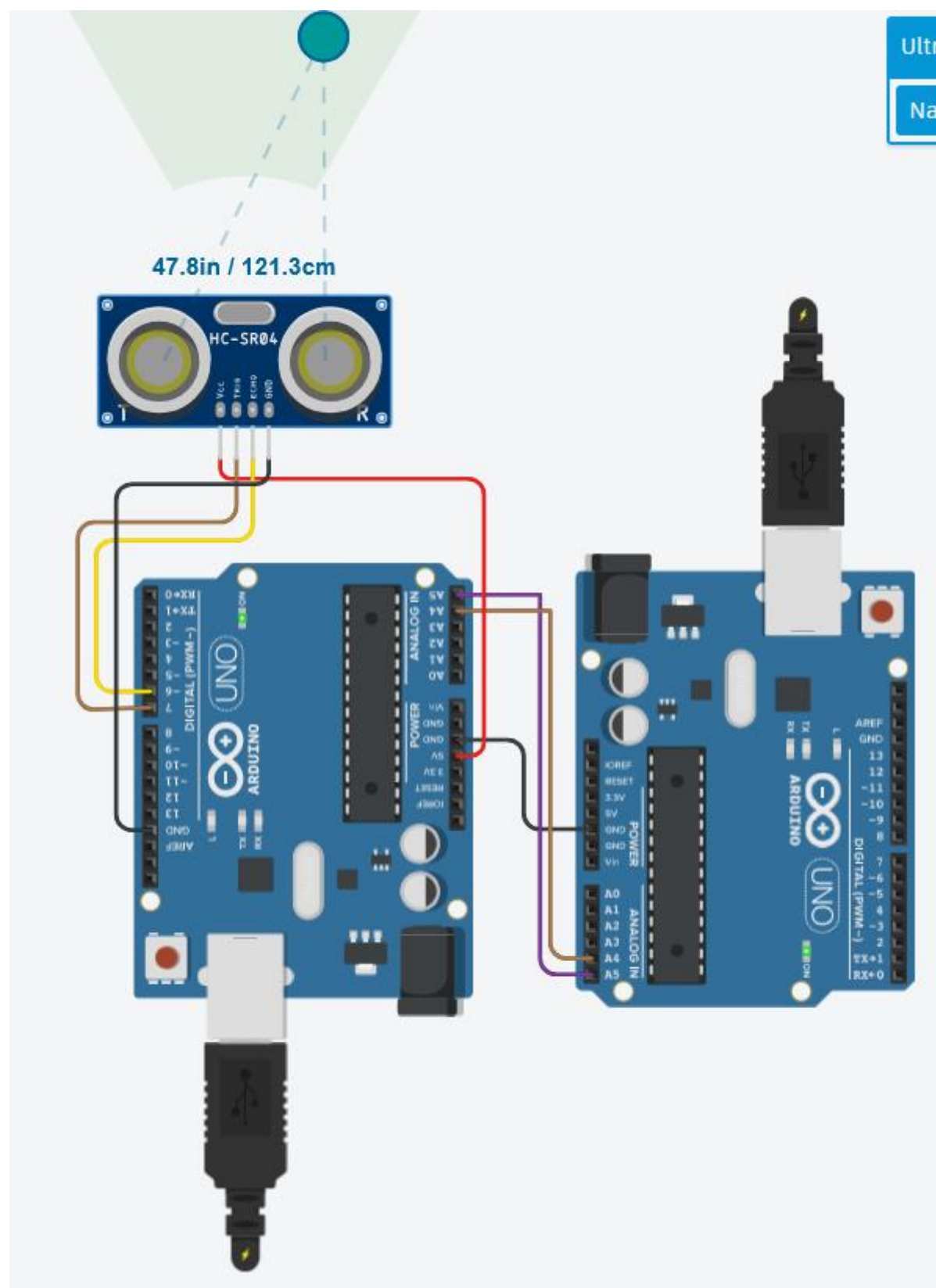


Fig. 2.4. Circuitul pornit

## Concluzie

În concluzie, cele două fragmente de cod Arduino prezintă roluri contrastante pentru aceeași placă de microcontroler. (Maestru): Acest cod configurează placa ca master I2C, solicitând în mod activ date la distanță de la un alt microcontroler (MCU1). Acesta recuperează datele în octeți, le combină pentru a obține valoarea completă și o afișează pe monitorul serial. (Slave): Acest cod setează placa ca sclav I2C, răspunzând la solicitări de la un dispozitiv master (MCU1). Acesta utilizează un senzor cu ultrasunete pentru a măsura distanța. La primirea unei solicitări, aceasta măsoară distanța, stochează o parte din date într-un buffer și transmite aceste date înapoi către master prin I2C. Împreună, aceste coduri demonstrează modul în care comunicarea I2C permite microcontrolerelor să funcționeze ca un sistem, unul solicitând date, iar celălalt furnizând măsurători ale senzorului.

## ANEXA

```
#include <Wire.h>

#define SLAVE_ADDRESS 0x08 // I2C address of MCU1
#define BUFFER_SIZE 2 // Size of the data buffer

uint8_t buffer[BUFFER_SIZE]; // received data

void setup() {
    Serial.begin(9600);
    Wire.begin();
}

void loop() {
    requestDistance();
    delay(500);
}

// request distance data
void requestDistance() {
    Wire.requestFrom(SLAVE_ADDRESS, BUFFER_SIZE);
    while (Wire.available() < BUFFER_SIZE); // Wait until data is received
    for (int i = 0; i < BUFFER_SIZE; i++) {
        buffer[i] = Wire.read();
    }
    int distance = buffer[0] | (buffer[1] << 8);
    Serial.print("Distance: ");
    Serial.print(distance);
    Serial.println(" cm");
}
```



## **Bibliografie**

1. TINKERCAD: *Arduino Simulator*. Online Simulator, © 2024 Autodesk, Inc [21.02.2024], Link: <https://www.tinkercad.com/dashboard>

3. ARDUINO DOCS: *Arduino Documentation and Examples*. Documentation, Examples and Libraries, ©2024 Arduino [21.02.2024], Link: <https://docs.arduino.cc/built-in-examples/>