**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**RAPORT**

Lucrare de laborator nr. 5

la cursul ***„Interfețe de comunicare”***

**Tema: Cercetarea interfetelor standard de comunicatii in cod serie. Regim Master-Slave.**

**A efectuat :**  **St. gr. CR-221FR Serba Cristina**

**A verificat:**  **Ababii Victor**

**Chișinău 2024**

# Scopul lucrării:

# Cercetarea, proiectarea și programarea interfețelor standard de comunicare in cod serie (COM, RS, UART, USART, I2C, SPI ) - Regim Master-Slave. Proiectarea în mediul Proteus sau Fritzing a unui sistem de transfer date în cod serie în baza Kit-ului Arduino UNO, cu interactiune Master-Slave. Dispozitivul Master va genera apeluri catre Slave care va raspunde prin oferirea datelor solicitate. Datele comunicate dintre dispozitive sunt afisate pe Terminal si pe Osciloscop. Initierea uni apel Master este generat de tastarea unui buton. Datele receptionate de Master sunt afișate pe un indicator LCD 2\*16 pozitii LM016L. Datele generate de Slave sunt achizitionate de la intrarile Analogice A0-A5 sau de la un set de senzori analogici sau digitali.

# Sarcina tehnică:

# 1. Să se instaleze mediul de dezvoltare Arduino IDE și bibliotecile respective;

# 2. Să se instaleze mediul de dezvoltare Proteus și bibliotecile respective;

# 3. Să se asambleze schema electrică de principiu (Proteus) a sistemului pentru comunlcare in regim Master-Slave. Datele generate de Slave sunt achizitionate de la intrarile Analogice A0-A5 sau de la un set de senzori analogici sau digitali. Comunicarea dintre dispozitive este monitorizata de modulul Terminal si sunt afisate si pe Osciloscop. Dispozitivul Master va genera apeluri catre Slave care va raspunde prin oferirea datelor solicitate. Initierea unui apel Master este generat de tastarea unui buton. Datele receptionate de Master sunt afișate pe un indicator LCD 2\*16 pozitii LM016L.

# 4. Să se elaboreze pentru Dispozitivul Slave programul (Arduino IDE) care efectuează achiziția datelor de la setul de senzori sau intrarile A0-A5, si livrarea acestora in raport cu codul receptionat de la dispozitivul Master.

# 5. Să se elaboreze pentru Dispozitivul Master (Arduino UNO) care va genera comenzi la tastarea unui buton pentru Dispozitivul Slave, primeste datele de la Slave si le afiseaza pe un ecran LCD LM016L;

6. Să se incarce codul HEX al programului elaborat în mediul Proteus pentru

testare funcțională;

7. Procesul de simulare sa se înregistreze prin PrintScrean-uri.

# Schema electrică de principiu:

# Descrierea sarcinii în parametri tehnici:

Sarcina presupune conectarea a două board-uri Arduino, în mod Master și Slave, colecționarea datelor de către Slave de la niște potențiometre, iar afișarea acestora de către Master la un display LCD LM016L. Fiecare potențiometru este caracterizat de un buton, care la apăsare va trimite semnal Slave-ului prin Master, apoi datele respective vor fi afișate.

# Algoritmul de funcționare al sistemului

Între board-urile Arduino Master și Slave a fost creată conexiune UART, respectiv au fost conectate Rx-Tx și invers al fiecăruia.

Pentru a avea butoane corespunzătoare fiecărui potențiometru, au fost legate câte unul de acestea, apoi conectate la Ground, în această porțiune fiind colectat semnalul butonului apăsat.

Un osciloscop a fost adăugat la conexiunea UART pentru a monitoriza semnalele trimise/primite.

# Codul sursă C și HEX ale programelor elaborate:

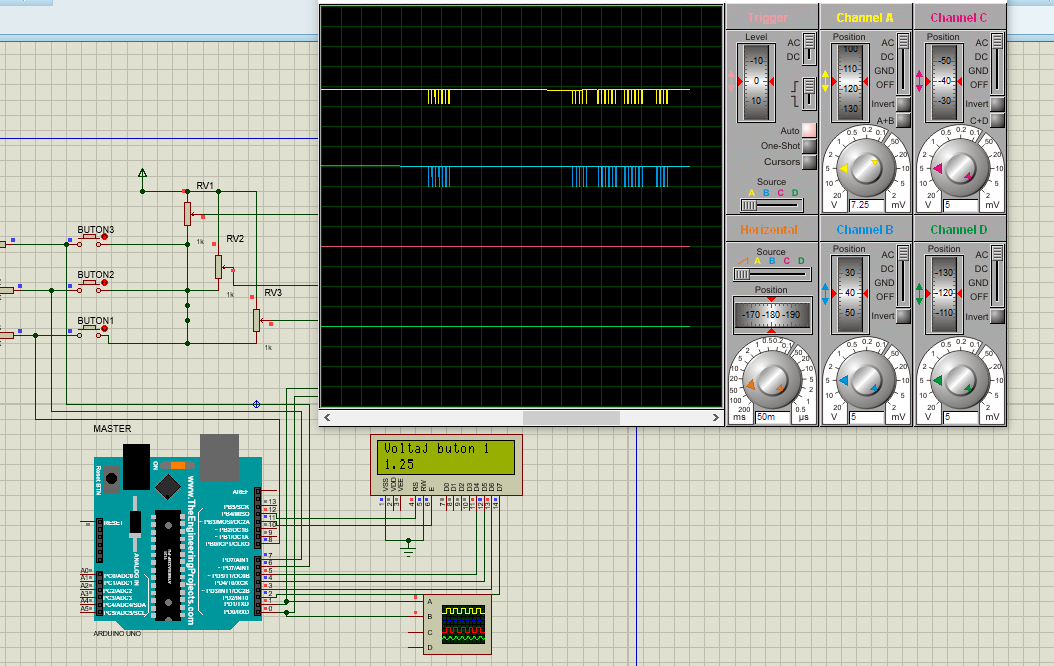
MASTER:

|  |
| --- |
| /\* Codul board-ului Master \*/  #include <LiquidCrystal.h>  #define NUM\_OF\_BUTTONS 3  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  int dataBuffer[NUM\_OF\_BUTTONS];  int pins[NUM\_OF\_BUTTONS];  void setup()  {    /\* Setarea pinilor pentru buton \*/    pins[0] = 8;    pins[1] = 7;    pins[2] = 6;    pinMode(pins[0], INPUT);    pinMode(pins[1], INPUT);    pinMode(pins[2], INPUT);    /\* Initializarea LCD-ului \*/    lcd.begin(16, 2);    /\* Initializarea comunicarii prin UART \*/    Serial.begin(9600);  }  void loop()  {    /\* Pentru fiecare buton \*/    for(int iter = 0u; iter < NUM\_OF\_BUTTONS; iter++)    {      /\* Verifica daca a fost apasat \*/      if (digitalRead(pins[iter]) == HIGH)      {        /\* Trimite semnal board-ului slave \*/        Serial.write(iter);        /\* Verifica daca a primit raspuns \*/        if (Serial.available() > 0)        {          lcd.clear(); /\* Curata ecranul \*/          int value = Serial.read(); /\* Citeste valoarea raw \*/          float convertedValue = (value \* 5.0) / 1023; /\* Formula de conversie a semnalului voltaj \*/          /\* Afisarea datelor primite \*/          lcd.setCursor(0, 0);          lcd.print("Voltaj ");          lcd.print(iter);          lcd.setCursor(0, 1);          lcd.print(convertedValue);        }      }    }    } |

SLAVE:

|  |
| --- |
| /\* Codul board-ului Slave \*/  #define NUM\_OF\_PINS 3  int pins[NUM\_OF\_PINS];  void setup()  {    pins[0] = A0;    pins[1] = A1;    pins[2] = A2;    Serial.begin(9600);  }  void loop()  {    /\* Verifica daca a primit semnal de la master \*/    if (Serial.available() > 0)    {      /\* In dependenta de indicele primit \*/      int cod = Serial.read();        /\* Citeste valoarea de la potentiometrul respectiv \*/      int value = analogRead(pins[cod]);        /\* Trimite datele citite inapoi la master \*/      Serial.write(value);    }  } |

# Graficele obţinute în rezultatul simulărilor / măsurărilor efectuate.



# Concluzii

În urma efectuării lucrării de laborator date, care avea ca scop familiarizarea cu interfața de comunicare UART în modul Master și Slave și lucrul cu senzori, am reușit să conectez cu succes o un Display LCD LM016L, două microcontrollere de tip Arduino Uno și potențiometre pentru colectarea semnalului.   
În concluzie, am acumulat cunoștințe noi despre comunicarea în mod master și slave și am reușit să integrez propriul circuit precum și am adâncit cunoștintele despre modul de lucru în Proteus și Arduino IDE. De asemenea, am putut monitoriza semnalele de intrare și ieșire prin conectarea unui osciloscop.