**Referat Tema 2**

**Analizor logic cu 8 canale**

|  |  |  |
| --- | --- | --- |
| **Echipa:** | Bolohan Marian-Cristian, Dodi Mircea-Ovidiu | 321AC |

**Cerințe temă:** Frecventa maxima a semnalului de ceas: 20 Mhz, Memorare 64000 secvente; Utilizare EEPROM I2C 256Kb

**Implementare cerințe temă**

1. **Schema bloc și Organigrama folosită pentru implementarea temei:**

**Schema bloc:**

**A blue circuit board with black wires

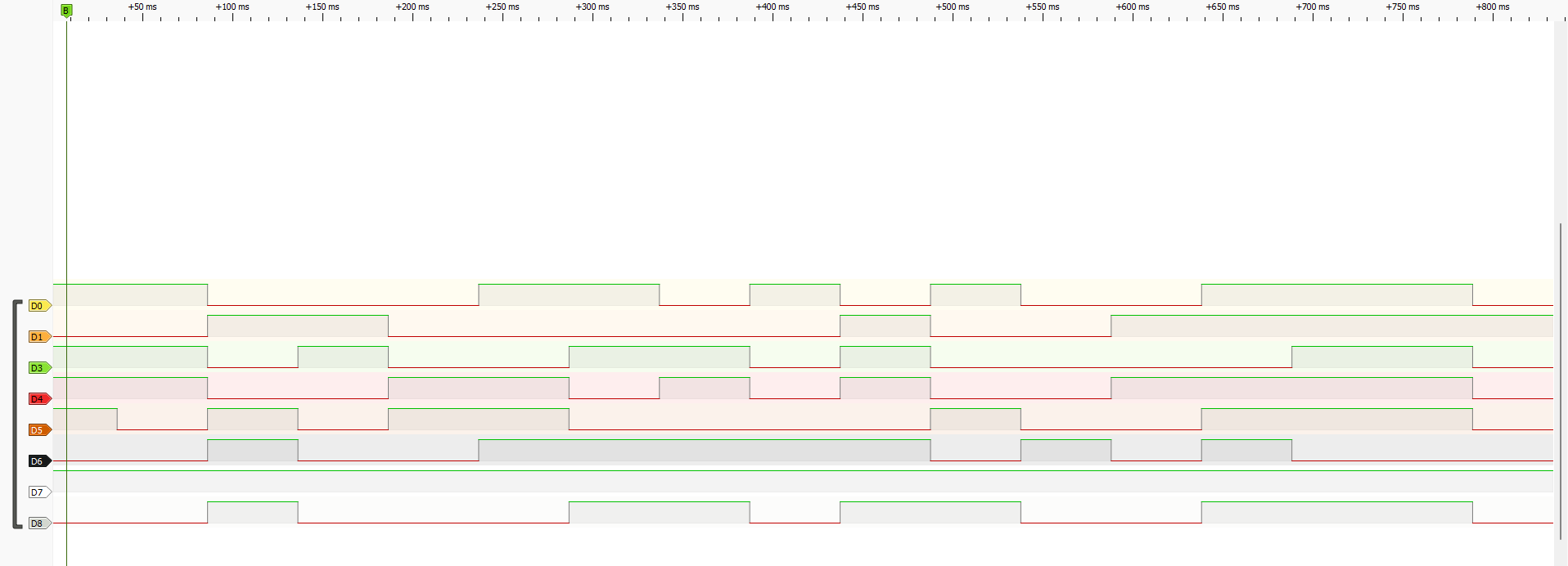
Description automatically generated**

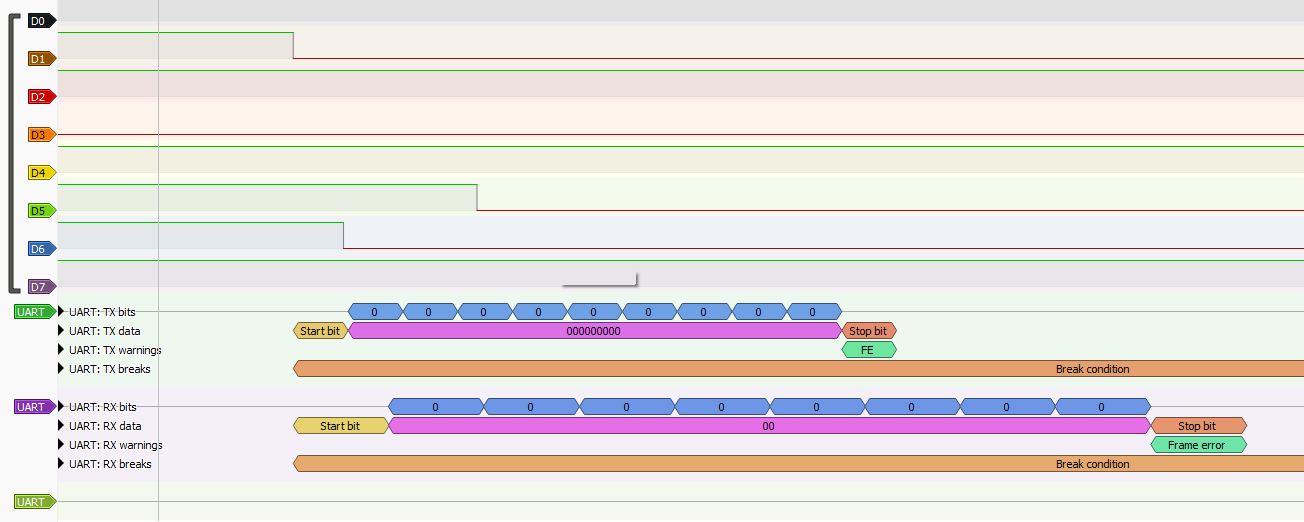
PC

**A.1. Modul Interfațare Grafică (GUI)**

Pentru modulul GUI, am folosit utilitarul open-source **PulseView**, conținut în **Sigrok**, un soft popular printre entuziaștii din domeniul electronicii.

Modului GUI afișează datele sub următoarea formă:





În imaginile precedente, frecvența utilizată este de 12Mhz și sunt prelevate 5M sample-uri. Pentru o analizare mai atentă, am pus în PulseView 3 interpetoare pentru protoculul UART (Universal Asynchronous Receiver/Transmitter), unde putem vedea la nivel de bit informația primită.

Pentru următoarele cerințe completăm cu secvențele relevante din codul rulat pe modului Arduino Uno. Dintre acestea, unele au rolul de a trimite date în loop(), iar altele de a primi de la loop() pentru afișare/memorare. Funcționalitatea propriu-zisă de afișare este îndeplinită de PulseView, care este configurat să citeasca datele de la analizorul pe 8 canale Saleae Pro, configurat cu ajutorul utilitarului **Zadig** din Sigrok.

**Date intrare (inițializare):**

A screenshot of a computer program

Description automatically generated

**Date transmise către Main:**

A computer code with white text

Description automatically generated

**Date primite de la Main pentru afișare:**

**A screenshot of a computer program

Description automatically generated**

**A.2. Modul Activitate Principală (Main)**

Funcțiile definite precedent sunt apelate în funcția loop(), ce joacă rolul funcției main a programului. Natura compliată a limbajului C++, pe care modulul Arduino este bazat, permite definirea acestor funcții după scrierea funcției main.

**A computer screen shot of a program

Description automatically generated**

1. **Participarea echipei la realizarea cerințelor temei:**
   1. **Student: Bolohan Marian-Cristian**

Participare la Realizare Modul:

**Schema bloc și Organigrama folosită pentru implementarea temei:**

Subrutină/e: configurarea analizorului Saleae Pro utilizând Zadig cu driverele necesare (WinUSB), asamblarea componentelor.

**Modul Activitate Principală (Main)**

Subrutină/e: Scrierea secvențelor de cod responsabile de scriere și memorare, optimizarea codului.

* 1. **Student: Dodi Mircea-Ovidiu**

Participare la Realizare Modul:

**Schema bloc și Organigrama folosită pentru implementarea temei:**

Subrutină/e**:** Asamblarea componentelor, scrierea secvențelor de cod responsabile de emiterea semnalelor sub formă de bytes, apelarea funcțiilor de cod în main.

**Modul Interfațare Grafică (GUI):**

Subrutină/e: configurare Sigrok, PulseView.

1. **Denumire fișiere utilizate pentru implementarea temei:**

**C.1. Modul Interfațare Grafică (GUI):**

Pentru implementarea interfeței grafice am folosit **PulseView**, soft component din utilitarul open-source **Sigrok**, disponibil la pagina sa oficială: <https://sigrok.org/wiki/Main_Page>.

**C.2. Modul Activitate Principală (Main):**

Codul rulat pe placuța Arduino Uno, ce conține modului Main se află în fișierul 8ch\_analyzer.ino