



INSTITUTUL DE  
ȘTIINȚE SPAȚIALE  
Filială INFLPR

Str. Atomiștilor, 409, Măgurele, Jud. Ilfov,  
O.P. MG-23, RO 077125, România

## **RAPORTUL ȘTIINȚIFIC ȘI TEHNIC (RST)**

### **ETAPA DE EXECUȚIE NR. 3**

**CU TITLUL “Integrarea capului de măsură al instrumentului HBPSKP – Partea I”**

### **PRECIZĂRI PRIVIND STRUCTURA RAPORTULUI ȘTIINȚIFIC ȘI TEHNIC**

#### **Cuprinde:**

**A.** Rezumatul etapei

**B.** Descrierea științifică și tehnică, cu punerea în evidență a rezultatelor etapei și gradul de realizare a obiectivelor (se indică rezultatele și modul de diseminare a rezultatelor)

**C.** Concluzii

**D.** Bibliografie

**Obiectivul general** al prezentei etape de execuție a implicat realizarea de activități pentru integrarea capului de măsură al instrumentului HBPSKP.

**Obiectivele etapei de execuție** au inclus realizarea activităților privind dezvoltarea în fază primară a modului electronic de control a capului de citire al instrumentului HBPSKP, precum și realizarea activităților necesare dezvoltării aplicației software de control și analiză de date pentru capul de măsură al instrumentului HBPSKP, fază finală.

## **A. Rezumatul etapei**

În conformitate cu Planul de Realizare al proiectului, în cadrul acestei etape au fost realizate activități necesare integrării capului de măsură al instrumentului HBPSKP. Au fost efectuate operațiile privind dezvoltarea în fază primară a modului electronic de control a capului de citire al instrumentului HBPSKP (Partea I). De asemenea, au fost realizate și operațiile pentru dezvoltarea aplicației software de control și analiză de date a capului de măsură al instrumentului HBPSKP (Partea II). Aplicația software de control și analiză de date a capului de măsură al instrumentului HBPSKP a fost elaborată în două variante: una, dezvoltată în limbajul de programare C/C++ sub mediul integrat Microsoft Visual Studio Community 2015 și o a doua sub mediul integrat LabView.

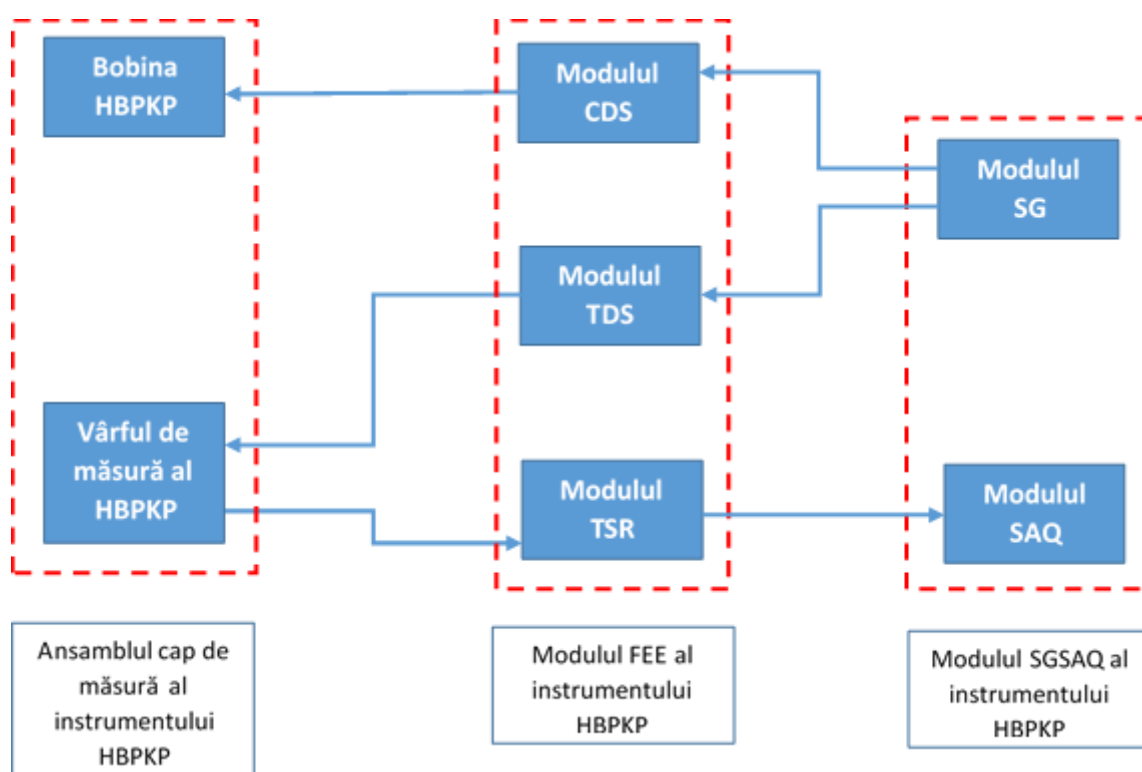
## **B. Descrierea științifică și tehnică, cu punerea în evidență a rezultatelor etapei și gradul de realizare a obiectivelor (se indică rezultatele și modul de diseminare a rezultatelor)**

**Activitatea realizată:** a constat din efectuarea operațiilor privind dezvoltarea în fază primară a modului electronic de control a capului de citire al instrumentului HBPSKP (Partea I), subsumate celei de a treia etape de execuție – **activitatea 3.1 din planul de realizare**.

În conformitate cu Planul de Realizare al proiectului, în cadrul acestei activități a fost proiectat și a fost începută construcția modului electronic de control a capului de citire al instrumentului HBPSKP. Operațiunile implicate de realizarea acestei activități au fost efectuate în colaborare, de către cei trei parteneri ai proiectului (Institutul de Științe Spațiale – **ISS**, Institutul Național de Cercetare-Dezvoltare pentru Mecanică Fină – **INCDFM** și **OPTOELECTRONICA 2001 S.A. – OPTO**).

Arhitectura modului electronic front-end (**FEE**) este prezentată în Figura 1 de mai jos, iar funcționarea sa poate fi descrisă după cum urmează. (Sub)Modulul SG (Signal Generation) al modului SGSAQ (Signal Generation and Signal Acquisition) – în fapt modulul de output al plăcii de achiziție de date NI USB-6351 – generează două semnale sinusoidale de amplitudini, frecvențe și faze diferite. Unul dintre aceste semnale va fi transmis către (sub)modulul CDS al modului FEE, unde va fi preprocesat și apoi aplicat pe bobina actuatorului magnetic pentru a produce oscilația mecanică a vârfului capului de măsură al instrumentului HBPKP.

Al doilea semnal – care în fapt este chiar semnalul de backing al vârfului instrumentului HBPKP – va fi transmis către (sub)modulul TDS (Tip Driving Signal) al modulului FEE, unde va fi preprocesat și apoi aplicat direct pe vârful capului de măsură al instrumentului. Semnalul produs de vârful capului de măsură al instrumentului ca urmare a oscilației sale mecanice combinată cu semnalul de backing aplicat va fi preluat de (sub)modulul TSR (Tip Signal Readout) al modulului FEE, preprocesat și transmis (sub)modulului SAQ (Signal Acquisition) al modulului SGSAQ – în fapt modulul de input al plăcii de achiziție de date NI USB-6351 – pentru achiziție și procesare ulterioară în vederea asigurării operației continue a instrumentului.



**Figura 1.** Schema bloc a instrumentului HBPSKP

Din punct de vedere practic, arhitectura modulului FEE este implementată pe o singură placă de circuit, iar transmiterea semnalelor către și de la această placă la restul sistemelor se face cu ajutorul cablurilor coaxiale pentru a minimiza efectele zgomotului electronic asupra funcționării instrumentului. Schemele electronice ale modulului FEE au fost realizate utilizând pachetul de software Altium Designer, și sunt prezentate pentru fiecare modul în Anexele 1, 2 și 3 ale prezentului raport.

### **Modulul CDS al modulului FEE**

Schema electronică a acestui modul este prezentată în Anexa 1. Așa cum se menționa în secțiunea precedentă, rolul acestui modul este acela de a interfața modulul de output al plăcii de achiziție de date NI USB-6531 cu bobina actuatorului mecanic al oscilatorului mecanic al vârfului capului de măsură al instrumentului HBPKP, care necesită puteri mari (e.g. 4 V la 750 mA) relativ la puterea de output a semnalelor generate de placă (e.g. 4V la maximum 5 mA). În aceste condiții, semnalul generat de canalul A0 al modulului de output al plăcii de achiziție de date este transmis cu ajutorul unui cablu coaxial RG58 la conectorul BNC P1 și este preluat ca input de către modulul CDS. Deoarece modulul de output al plăcii de achiziție de date are o referință separată de semnal (un "pământ" separat) față de referința de semnal a capului de măsură al instrumentului HBPKP, și deoarece chiar și în cazul utilizării de cabluri coaxiale pentru transmiterea semnalelor între diferitele module ale instrumentului aceste semnale pot prelua zgomot electronic din mediul electromagnetic înconjurător atât pe firul de semnal cât și pe tresa de ecranare a cablului coaxial, semnalul generat de canalul A0 al modulului de output al plăcii de achiziție de date este preluat de modulul CDS printr-un etaj de intrare diferențial. Acest etaj de intrare diferențial utilizează un opamp (amplificator operațional) de precizie AD713JN în configurație de diferențiator (U1A în schema din Anexa 1) și are următoarele trei roluri:

1. În primul rând, ca diferențiator, acest etaj preia semnalul de la canalul A0 al modulului de output al plăcii de achiziție de date (cu referința de pământ a plăcii) și îl transformă într-un semnal cu referința de pământ GND (vezi schema din Anexa 1), care este referința de semnal pentru electronica întregului cap de măsură al instrumentului HBPKP. În același timp, configurația diferențială contribuie și la reducerea zgomotului electronic (common mode noise) preluat din mediul electromagnetic înconjurător de cablul coaxial de transmisie a semnalului.
2. În al doilea rând, utilizând performanța de CMRR (Common Mode Rejection Ratio) a opamp-ului, se elimină cea mai mare parte a zgomotului electronic (common mode noise) preluat din mediul electromagnetic înconjurător de cablul coaxial de transmisie.
3. În afara de funcția de diferențiere, etajul mai are și o funcție de filtrare de semnal, limitând lărgimea de bandă a semnalului de intrare la ~10 kHz pentru a elimina orice tip de zgomot asociat cu frecvențe înalte și nenesare operării instrumentului.

Etajului de input îi urmează un etaj de output, care folosește un opamp de putere de tip OPA547T (U2 în schema din Anexa 1) în configurație de booster buffer (follower) pentru a furniza tensiunea și curentul necesare pentru operarea bobinei oscilatorului mecanic al vârfului capului de măsură al instrumentului HBPKP. Pentru a preveni distrugerea accidentală

a acestei bobine, utilizând caracteristicile constructive ale acestui opamp, curentul de output este limitat la ~400 mA cu ajutorul unei rezistențe de 30 k $\Omega$  conectate între pinii 3 și 4 ai acestui opamp, în condițiile în care curentul maxim prin bobină este de 750 mA. Outputul acestui etaj este conectat direct la bobina oscilatorului mecanic, și pentru a preveni preluarea de zgomot electronic din mediul electromagnetic înconjurător, întreg modulul FEE va fi montat fizic foarte aproape de bobina oscilatorului mecanic (aproape de capul de măsură în general) pentru a putea folosi fire foarte scurte.

Acest modul, ca de altfel întregul modul FEE, este alimentat de o sursă de tensiune în comutare de tip MeanWell PT-65B de 65W, care generează tensiunile de -12V/+12V necesare pentru operarea tuturor opamp-urilor și componentelor electronice ale modulului.

### **Modulul TDS al modulului FEE**

Schema electronică a acestui modul este prezentată în Anexa 2. Conform celor menționate anterior, rolul acestui modul este acela de a interfața modulul de output al plăcii de achiziție de date NI USB-6531 cu vârful capului de măsură al instrumentului HBPKP. În aceste condiții, semnalul generat de canalul A1 al modulului de output al plăcii de achiziție de date este transmis cu ajutorul unui cablu coaxial RG58 la conectorul BNC P2 și este preluat ca input de către modulul TDS. Deoarece modulul de output al plăcii de achiziție de date are o referință separată de semnal (un "pământ" separat) față de referința de semnal a capului de măsură al instrumentului HBPKP, și deoarece chiar și în cazul utilizării de cabluri coaxiale pentru transmiterea semnalelor între diferitele module ale instrumentului aceste semnale pot prelua zgomot electronic din mediul electromagnetic înconjurător atât pe firul de semnal cât și pe tresa de ecranare a cablului coaxial, semnalul generat de canalul A1 al modulului de output al plăcii de achiziție de date este preluat de modulul TDS printr-un etaj de intrare diferențial. Acest etaj de intrare diferențial utilizează un opamp (amplificator operațional) de precizie AD713JN în configurație de diferențiator (U1C în schema din Anexa 2) și are următoarele trei roluri:

1. În primul rând, ca diferențiator, acest etaj preia semnalul de la canalul A1 al modulului de output al plăcii de achiziție de date (cu referința de pământ a plăcii) și îl transformă într-un semnal cu referința de pământ GND (vezi schema din Anexa 2), care este referința de semnal pentru electronica întregului cap de măsură al instrumentului HBPKP. În același timp, configurația diferențială contribuie și la reducerea zgomotului electronic (common mode noise) preluat din mediul electromagnetic înconjurător de cablul coaxial de transmisie a semnalului.
2. În al doilea rând, utilizând performanța de CMRR (Common Mode Rejection Ratio) a opamp-ului, se elimină cea mai mare parte a zgomotului electronic (common mode

noise) preluat din mediul electromagnetic înconjurător de cablul coaxial de transmisie.

3. În afara de funcția de diferențiere, etajul mai are și o funcție de filtrare de semnal, limitând lărgimea de bandă a semnalului de intrare la ~1 kHz pentru a elimina orice tip de zgomot asociat cu frecvențe înalte și nenesare operării instrumentului.

Outputul acestui etaj, TDS\_Out, care reprezintă de fapt potențialul armonic/sinusoidal de backing, este preluat ca input de modulul TSR al modulului FEE, și este aplicat indirect de către acesta din urmă pe vârful capului de măsură al instrumentului. Și acest modul, ca de altfel și modulul CDS, este alimentat la sursa de tensiune în comutare MeanWell PT-65B de 65W, care generează tensiunile de -12V/+12V necesare pentru operarea tuturor opamp-urilor și componentelor electronice ale modulului FEE.

### **Modulul TSR al modulului FEE**

Schema electronică a acestui modul este prezentată în Anexa 3. Conform celor menționate anterior, rolul acestui modul este acela de a interfața vârful capului de măsură al instrumentului HBPKP cu modulul de input al plăcii de achiziție de date NI USB-6531. În aceste condiții, etajul de input al modulului constă într-un convertor curent-tensiune format dintr-un opamp de precizie AD713JN (U1B în schema din Anexa 3) care preia ca input de la modulul TDS potențialul de backing TDS\_Out pe care îl aplică indirect vârfului cu ajutorul intrării sale ne-invertoare, și de la vârful instrumentului semnalul de curent generat între vârf și probă (semnalul "de măsură" al instrumentului) pe care îl convertește într-un semnal de tensiune cu ajutorul unei transimpedanțe de 1 MΩ.

Acesta din urmă este preluat de al doilea etaj al modulului, care utilizează un opamp de precizie AD713JN (U1D în schema din Anexa 3) pentru implementarea unui filtru cu lărgimea de bandă de 10 kHz al cărui rol este acela de a elimina zgomotul preluat de semnalul de vârf din mediul electromagnetic ambiant pe firele de legătura dintre vârf și etajul de intrare al modulului TSR.

Outputul etajului de filtrare – care în fapt reprezintă outputul modulului TSR este transmis către conectorul P3, și transmis printr-un cablu coaxial de tip RG 58 către modulul de input al plăcii de achiziție de date NI USB-6531, unde este preluat diferențial de canalul A0 de input al plăcii. Din nou este utilizată tehnica preluării diferențiale de către placa de achiziție de date a semnalului de output a modulului TSR pentru a schimba referința semnalului de la pământul capului de măsură al instrumentului (GND în schema din Anexa 3) la pământul plăcii de achiziție de date și pentru a elimina zgomotul electronic de tip common mode preluat pe cablul de transmisie. Și acest modul, ca de altfel și modulele CDS și TDS, este alimentat la sursa de tensiune în comutare MeanWell PT-65B de 65W, care generează

tensiunile de -12V/+12V necesare pentru operarea tuturor opamp-urilor și componentelor electronice ale modului FEE.

**Schemele electrice ale (sub)modulelor componente ale modului FEE au fost verificate cu ajutorul pachetului de software Altium Designer, și în prezent electronica modului FEE este în curs de fabricație.**

**Activitatea realizată:** a constat din efectuarea operațiilor pentru dezvoltarea aplicației software de control și analiză de date a capului de măsură al instrumentului HBPSKP (Partea II) – **activitatea 3.2 din planul de realizare.**

Operațiunile implicate de realizarea acestei activități au fost efectuate în colaborare de către cei trei parteneri ai proiectului (Institutul de Științe Spațiale – **ISS**, Institutul Național de Cercetare-Dezvoltare pentru Mecanică Fină – **INCDMF** și OPTOELECTRONICA 2001 S.A. – **OPTO**).

În conformitate cu Planul de Realizare al proiectului, în cadrul acestei activități a fost definitivată aplicația software de control și analiză de date pentru capul de măsură al instrumentului HBPSKP.

Eforturile depuse în cadrul etapei anterioare -2- pentru elaborarea unei aplicații software de control și analiză de date a capului de măsură al instrumentului HBPSKP, au fost continuate și în cadrul prezentei etape. Rezultatul obținut a fost o aplicație software complet operațională, dezvoltată în limbajul de programare C/C++ sub mediul integrat Microsoft Visual Studio Community 2015 [1].

Visual Studio Community 2015 este un mediu integrat de programare (**IDE**) gratuit și complet elaborat, pentru dezvoltatori individuali, proiecte „open-source”, cercetare academică, educație și mici echipe de profesioniști [2]. Cu ajutorul acestui mediu de programare se pot construi aplicații pentru sistemele de operare Windows, Android și iOS, precum și aplicații Web și servicii Cloud. Visual Studio Community 2015 include secțiuni de proiectare, editoare de text, depanatoare și profilere, toate într-un singur instrument.

Această aplicație software este inclusă în anexa 4 și conține două task-uri. Schema ei bloc este prezentată în figura 2, mai jos.

În cadrul primului task sunt aplicate două semnale de ieșire pe circuitele de output ale plăcii de achiziție National Instruments NI USB – 6351. Aceste semnale sunt de tip sinusoidal și au frecvențe și amplitudini diferite.

În cadrul celui de-al doilea task, semnalul electric (1-canal) furnizat de un generator de semnal cu două canale este achiziționat prin intermediul aceleiași plăci de achiziție NI USB – 6351. Amplitudinile măsurate sunt prelucrate matematic și constituie informația utilizată pentru update-ul amplitudinilor celor două canale de ieșire menționate mai sus.

Odată ce aceste operațiuni sunt realizate, ciclicitatea procedurii de scriere / citire este reluată atât timp cât utilizatorul dorește (până când este trimis semnalul de STOP Program de către user).

Aplicația software este structurată pe mai multe module (figura 2), ale căror sarcini sunt prezentate în detaliu în cele ce urmează:

## 1. Modulul de inițiere și definire date de intrare

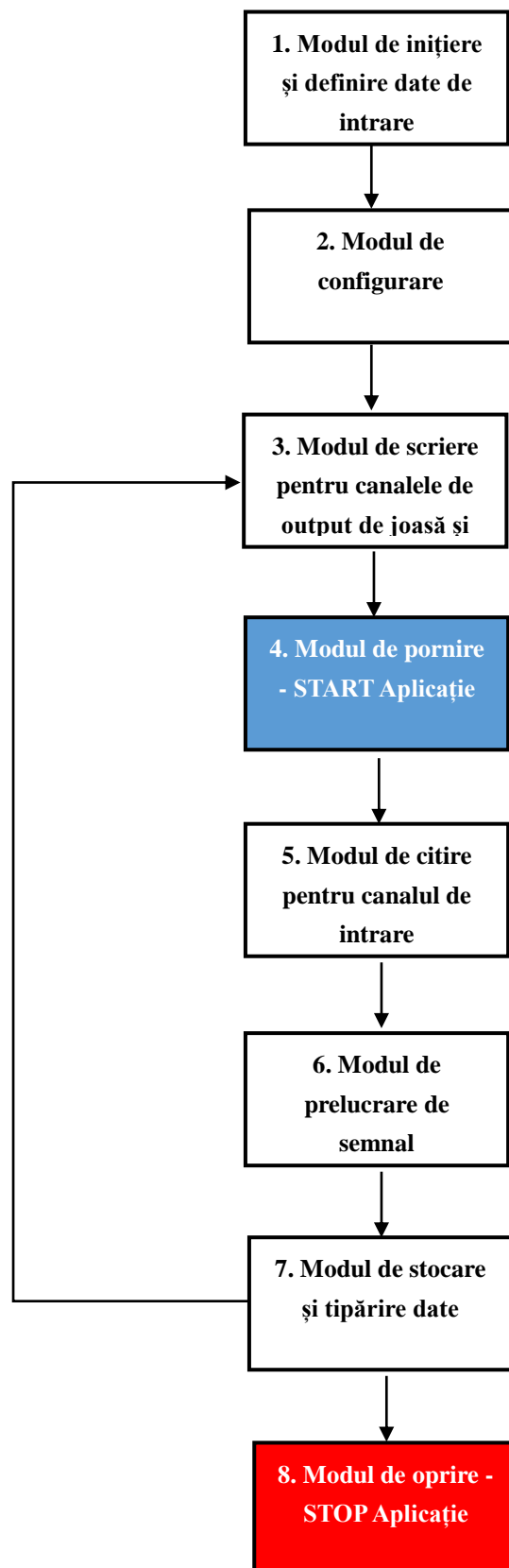
În cadrul acestui modul sunt definite datele inițiale ale aplicației software. În particular, aici sunt:

- incluse headerele standard necesare compilării aplicației software;
- definite constante globale (ex.  $\pi$ );
- definite și inițializate variabile locale pentru accesarea următorilor parametri:
  - rata de eșantionare pentru canalele de ieșire,
  - vectorul care conține valorile eșantioanelor pentru canalele de joasă și înaltă frecvență ce urmează a fi scrise,
  - numărul de eșantioane ce urmează a fi scrise pentru canalele de ieșire,
  - valorile maximă și minimă acceptate pentru amplitudinile semnalelor sinusoidale de ieșire,
  - timpul maxim de așteptare pentru scrierea tuturor eșantioanelor, în cazul canalelor de ieșire,
  - numărul de eșantioane pe ciclu ce urmează a fi scrise pentru canalul de ieșire de joasă, respectiv înaltă frecvență,
  - valoarea frecvenței pentru canalul de ieșire de joasă, respectiv înaltă frecvență,
  - valorile amplitudinilor inițiale pentru semnalele sinusoidale ale canalelor de ieșire de joasă, respectiv înaltă frecvență,
  - vectorii care conțin, separat, valorile eșantioanelor de joasă, respectiv de înaltă frecvență ce urmează a fi scrise,
  - rata de eșantionare pentru canalul de intrare,
  - numărul de eșantioane ce urmează a fi achiziționat pentru canalul de intrare,
  - valorile maximă și minimă acceptate pentru amplitudinea semnalului sinusoidal de intrare,
  - timpul maxim de așteptare pentru scrierea tuturor eșantioanelor, în cazul canalului de intrare,
  - vectorul care conține valorile eșantioanelor, ce urmează a fi citite, pentru canalul de intrare.



## 2. Modulul de configurare

În cadrul acestui modul sunt configurate caracteristicile electrice ale semnalelor ce



**Figura 2.** Schema bloc a aplicației software de control și analiză de date pentru capul de măsură al instrumentului HBPSKP.

urmează a fi generate sau achiziționate. Această configurare se realizează prin apelarea în ordinea prezentată mai jos, a următoarelor funcții specifice driver-ului NI-DAQmx:

- ***int32 DAQmxCreateTask (const char taskName[], TaskHandle \*taskHandle)***. Această funcție este apelată de două ori. O dată pentru canalele de scriere (output) și a doua oară pentru canalul de citire (input). Conform [3] această funcție creează un task și impune, în final, utilizarea funcției ***DAQmxClearTask*** pentru eliminarea (încheierea) aceluși task la momentul final. Parametrii acestei funcții sunt: numele taskului și o referință către taskul creat în cadrul funcției. Valoarea întoarsă de funcție este codul de eroare care apare dacă s-a produs o eroare sau un avertisment. O valoare de „0” indică succesul;
  
- ***int32 DAQmxCreateAOVoltageChan (TaskHandle taskHandle, const char physicalChannel[], const char nameToAssignToChannel[], float64 minVal, float64 maxVal, int32 units, const char customScaleName[])***. Această funcție este apelată o dată în cazul task-ului asociat canalelor de scriere. Conform [4] această funcție creează canalul/canalele prin care se va genera tensiunea electrică și adaugă aceste canale task-ului creat anterior. Parametrii acestei funcții sunt: task-ul creat anterior la care se adaugă canalele de ieșire, numele (sau domeniul) canalelor fizice utilizate pentru a se crea canale virtuale, numele ce pot fi asociate canalelor virtuale, limitele minimă, respectiv maximă între care ne așteptăm să generăm semnal, unitatea de măsură în care se furnizează semnalele și numele unei scale proprii (adesea asignat cu NULL). Valoarea întoarsă de funcție este codul de eroare care apare dacă s-a produs o eroare sau un avertisment. O valoare de „0” indică succesul;
  
- ***int32 DAQmxCfgSampClkTiming (TaskHandle taskHandle, const char source[], float64 rate, int32 activeEdge, int32 sampleMode, uInt64 sampsPerChanToAcquire)***. Această funcție este apelată de două ori. O dată pentru canalele de scriere (output) și a doua oară pentru canalul de citire (input). Conform [5] această funcție setează sursa ceasului de eșantionare, rata acestui ceas și numărul de eșantioane de achiziționat sau generat. Parametrii acestei funcții sunt: task-ul creat anterior la care se adaugă canalele de ieșire, respectiv canalul de intrare, sursa ceasului de eșantionare (NULL dacă se folosește ceasul intern al dispozitivului), rata de eșantionare în eșantioane / secundă / canal, specificarea marginii active a ceasului pentru achiziția sau generarea semnalelor (DAQmx\_Val\_Rising sau DAQmx\_Val\_Falling), specificarea modului de achiziție al eșantioanelor (continuu sau printr-un număr finit de eșantioane - DAQmx\_Val\_ContSamps sau DAQmx\_Val\_FiniteSamps) și numărul de eșantioane de achiziționat sau generat pentru fiecare canal din task. Valoarea întoarsă

de funcție este codul de eroare care apare dacă s-a produs o eroare sau un avertisment. O valoare de „0” indică succesul;

Tot în cadrul acestui modul de configurare, sunt citite de la consolă datele ce trebuie furnizate de către utilizator. Acestea sunt: frecvența canalului de ieșire de joasă frecvență și frecvența canalului de ieșire de înaltă frecvență.

### **3. Modulul de scriere pentru canalele de output de joasă, respectiv înaltă frecvență**

În cadrul acestui modul sunt generate valorile semnalelor de output pentru canalele de joasă, respectiv de înaltă frecvență. Generarea acestor date se face în interiorul unor bucle de tip „for”, în cadrul cărora se stochează valorile provenite de la cele două funcții tip sinusoidă, în doi vectori aferenți celor două tipuri de frecvențe. În cadrul unui al treilea ciclu, datele corespunzătoare celor două frecvențe mai sus amintite sunt inserate în cadrul unui singur vector cu dimensiunea dublă față de cei doi vectori anteriori. Se apelează apoi următoarea funcție specifică driver-ului NI-DAQmx:

***int32 DAQmxWriteAnalogF64 (TaskHandle taskHandle, int32 numSampsPerChan, bool32 autoStart, float64 timeout, bool32 dataLayout, float64 writeArray[], int32 \*sampsPerChanWritten, bool32 \*reserved).***

Această funcție este apelată o singură dată pentru ambele canale de output și conform [6] ea realizează scrierea unui număr de eșantioane de valoare reală în cadrul unui task care conține unul sau mai multe canale analoge de ieșire (în cazul de față – 2 astfel de canale). Parametrii acestei funcții sunt: task-ul creat anterior, numărul de eșantioane / canal ce vor fi scrise, un steag care semnalizează programului să pornească automat această funcție în cazul în care utilizatorul nu o face, cantitatea de timp de așteptare necesar scrierii tuturor eșantioanelor, modul de scriere a eșantioanelor (întrețesut « DAQmx\_Val\_GroupByScanNumber » sau neîntrețesut « DAQmx\_Val\_GroupByChannel »), vectorul ce conține valorile celor două semnale care vor fi scrise, numărul de eșantioane scrise cu succes în buffer și un parametru rezervat de dezvoltator pentru aplicații viitoare. Valoarea întoarsă de funcție este codul de eroare care apare dacă s-a produs o eroare sau un avertisment. O valoare de „0” indică succesul.

### **4. Modulul de pornire – START aplicație**

În cadrul acestui modul sunt inițiate procedurile de scriere și citire a celor două semnale de output și semnalului de input. Pentru a realiza aceste operațiuni este apelată următoarea funcție specifică driver-ului NI-DAQmx: ***int32 DAQmxStartTask (TaskHandle taskHandle).***

Conform [7] această funcție realizează pentru task-ul asociat, tranziția de la starea de „pregătit” la starea „activ, operațional”, stare care începe măsurarea și/sau generarea propriu-zisă.

Funcția conține un singur parametru și anume task-ul creat anterior. Apelarea acestei funcții se face de două ori. O dată pentru task-ul asociat operațiunilor de scriere și o dată pentru task-ul asociat operațiunii de citire.

Valoarea întoarsă de funcție este codul de eroare care apare dacă s-a produs o eroare sau un avertisment. O valoare de „0” indică succesul.

## 5. Modulul de citire canal de intrare

În cadrul acestui modul se realizează operația de citire/achiziție a semnalului de input, prin intermediul următoarei funcții specifice driver-ului NI-DAQmx:

*int32 DAQmxReadAnalogF64 (TaskHandle taskHandle, int32 numSampsPerChan, float64 timeout, bool32 fillMode, float64 readArray[], uInt32 arraySizeInSamps, int32 \*sampsPerChanRead, bool32 \*reserved).*

Această funcție este apelată o singură dată pentru unicul canal de input și conform [8] ea realizează citirea unui număr de eșantioane de valoare reală în cadrul unui task care conține unul sau mai multe canale analoge de intrare (în cazul de față un singur canal).

Parametrii acestei funcții sunt: task-ul creat anterior, numărul de eșantioane / canal ce vor fi citite, cantitatea de timp de așteptare necesar citirii tuturor eșantioanelor, modul de citire a eșantioanelor (întrețesut « DAQmx\_Val\_GroupByScanNumber » sau neîntrețesut « DAQmx\_Val\_GroupByChannel »), vectorul ce conține valorile semnalului care vor fi citite, dimensiunea acestui vector, numărul de eșantioane citite cu succes și un parametru rezervat de dezvoltator pentru aplicații viitoare. Valoarea întoarsă de funcție este codul de eroare care apare dacă s-a produs o eroare sau un avertisment. O valoare de „0” indică succesul.

## 6. Modulul de prelucrare a semnalului achiziționat (input)

În cadrul acestui modul se realizează operațiile de prelucrare a semnalului de input. Valorile citite și stocate în vectorul aferent funcției precedente sunt utilizate în cadrul unor algoritmi pentru calcularea valorilor mediate necesare modificării amplitudinilor sinusoidelor semnalelor de ieșire. Acești algoritmi implică utilizarea unor bucle de tip „for” și operații matematice standard.

După ce valorile mediate sunt calculate, amplitudinile semnalelor sinusoidale de output sunt modificate și ciclul se poate relua prin revenirea la modulul 3. Totuși revenirea la modulul 3 se va realiza doar după finalizarea operațiunilor asociate modulului următor.

## 7. Modulul de stocare și tipărire date

În cadrul acestui modul se realizează operațiile de scriere a datelor obținute în fișiere de tip text și eventuala afișare pe ecran a unor secvențe de date în vederea verificării bunei funcționări a aplicației software. Scrierea datelor se realizează cu funcții C/C++ standard. La finalizarea acestor operațiuni ciclul aplicației software se reia prin revenirea la modulul 3.

Aplicația software funcționează într-o buclă de tip „while”, între modulele 3 și 7 atât timp cât utilizatorul nu intervine prin acționarea unei taste specifice de la tastatura computerului. Dacă acea tastă este acționată, se intră în ultimul modul și anume 8, iar aplicația software se oprește.

## 8. Modulul de oprire – STOP aplicație

În cadrul acestui modul se realizează operațiile de oprire a aplicației software și de dealocare de memorie. Operațiunile respective sunt realizate prin intermediul următoarelor două funcții specifice driver-ului NI-DAQmx:

- *int32 DAQmxStopTask (TaskHandle taskHandle)*. Conform [9] această funcție oprește task-ul asociat și întoarce aplicația la starea în care se găsea înainte de apelarea funcției «DAQmxStartTask». Funcția conține un singur parametru și anume task-ul creat anterior.
- *int32 DAQmxClearTask (TaskHandle taskHandle)*. Conform [10] această funcție elimină task-ul asociat. Înainte de eliminarea task-ului această funcție abandonează task-ul și dacă este necesar dealocă orice resursă de memorie rezervată de acel task. Funcția conține un singur parametru și anume task-ul creat anterior.

Apelarea acestor două funcții aferente modulului 8 se face de două ori. O dată pentru task-ul asociat operațiunilor de scriere și o dată pentru task-ul asociat operațiunii de citire.

Valoarea întoarsă de fiecare din cele două funcții este codul de eroare care apare dacă s-a produs o eroare sau un avertisment. O valoare de „0” indică succesul.

**Aplicația software prezentată mai sus a fost testată și funcționează în parametrii proiectați.**

Totuși, pentru ușurința utilizării a fost dezvoltată o versiune a programului descris mai sus și sub mediul integrat **LabView** [11]. Sistemul de dezvoltare **LabVIEW** este unul foarte puternic atunci când se face referire la crearea aplicațiilor **DAQ**. **LabVIEW** include un set de

VI-uri care permit configurarea și achiziționarea datelor de la plăcile **DAQ**, precum și trimiterea datelor la aceste dispozitive.

O singură placă de achiziție poate efectua o varietate de funcții, cum ar fi:

- conversia analog-digitală (**A/D**);
- conversia digital-analogică (**D/A**);
- operații intare-ieșire (**I/O**) digitale;
- operații de numărare/cronometrare.

Fiecare placă de achiziție **DAQ** acceptă diferite frecvențe de generare a semnalului. De asemenea, fiecare placă **DAQ** este proiectată pentru un anumit hardware și un anumit sistem de operare.

Scopul achiziției datelor este acela de a măsura un fenomen electric sau fizic, cum ar fi: tensiunea, temperatura, curentul, presiunea, sau sunetul.

### **Prezentarea aplicației software LabVIEW - DAQ USB 6351**

Această aplicație utilizează două module: modulul de achiziție semnal și modulul în care se generează două unde continue, de tip sinusoidal, utilizând frecvența de eșantionare internă a plăcii DAQ USB 6351. Regenerarea automată a datelor de ieșire de pe osciloscop, a fost dezactivată, astfel încât datele de ieșire să fie furnizate continuu pe toată durata operației de generare, către cele două porturi de ieșire. Aplicația permite generarea de frecvențe arbitrare deoarece VI-ul (Virtual Instrument) calculează noi date pentru fiecare iterație a buclei, menținând continuitatea fazei semnalelor.

### **Componente necesare:**

#### **Software**

- LabVIEW 2016 - mediul de dezvoltare;
- DAQmx driver pentru placa de achiziție DAQ USB 6351.

#### **Hardware**

- NI DAQ USB 6351 National Instruments.

### **Arhitectura de implementare software**

Diagrama bloc de implementare este prezentată în figura 3.

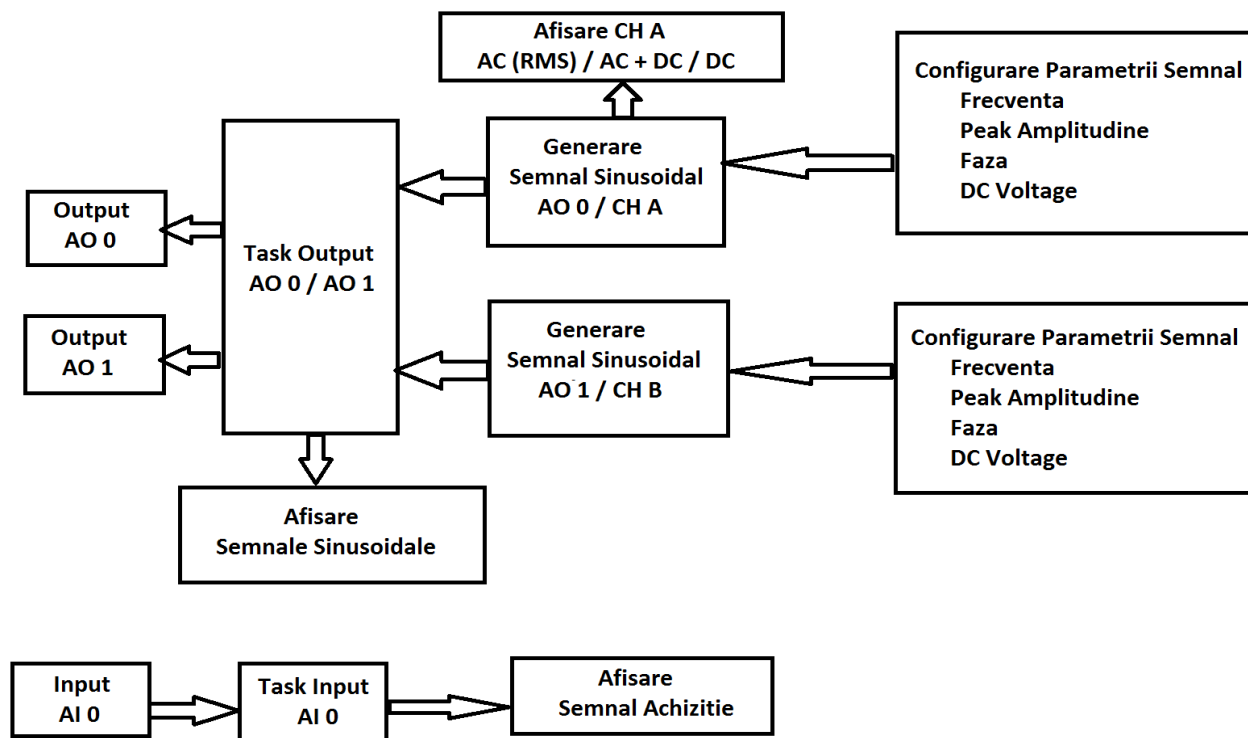


Figura 3. Diagrama bloc de implementare software.

### Etapele de implementare software

1. Se crează canalele de tensiune pentru intrarea / ieșirile analogice (DAQmx Create Channel vi).
2. Se configurează task-ul de stopare regenerare automată a datelor (Property Node DAQmx vi).
3. Se apelează DAQmx Sample Clock vi pentru a seta rata de eșantionare (DAQmx Timing Clock vi).
4. Citește rata reală a frecvenței de esantionare (sample clock) – funcție de specificațiile plăcii utilizate (Property Node DAQmx vi).
5. Se calculează forma de undă dorită, funcție de dimensiunea buffer-ului și valoarea frecvenței de eșantionare stabilită. Acest vi ține evidența fazei de undă pentru a se asigura că semnalul generat este continuu.
6. Se generează forma de undă spre buffer-ul de ieșire.
7. Se activează bucla continuă de generare (While Loop vi).
8. Bucla se generează continuu până când aplicația este oprită de către utilizator prin apăsarea butonului STOP. Fiecare iterație calculează și scrie o nouă formă de undă în buffer-ul de ieșire.
9. Se apelează funcția CLEAR vi pentru a șterge buffer-ul.
10. Apariția erorilor sunt afișate, prin vizualizarea unui dialog pop-up de afișare.

## Driver DAQmx

- Pentru utilizarea NI USB-6351 în LabVIEW se utilizează funcțiile DAQmx, conform figurii 4.

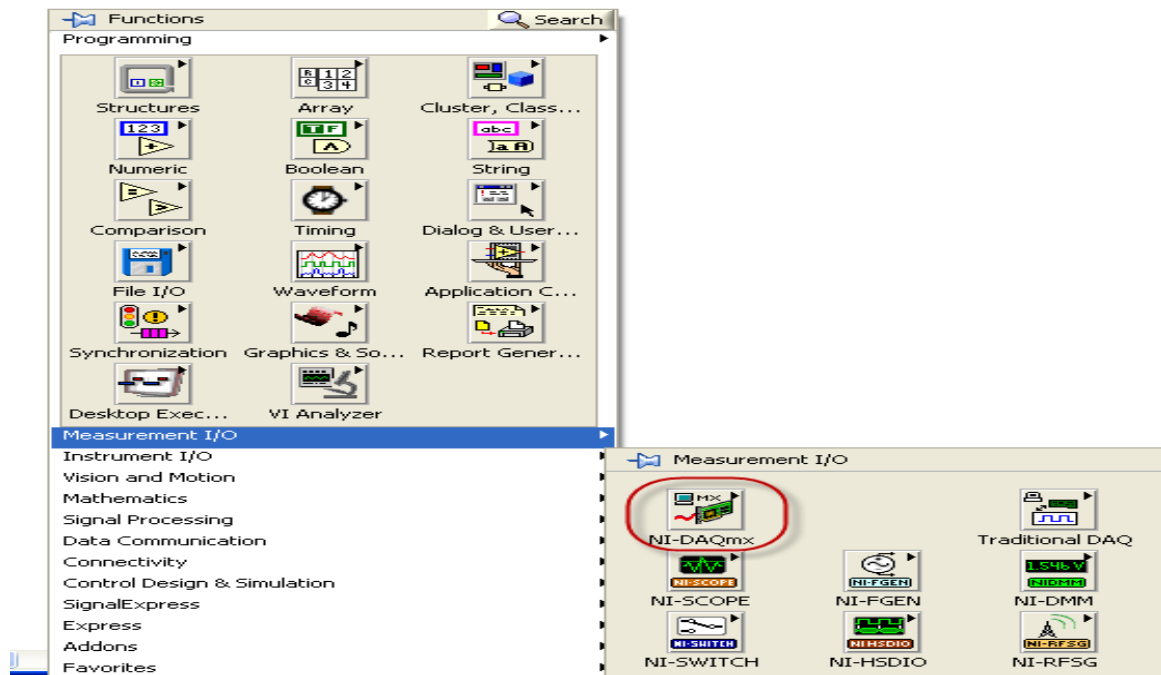
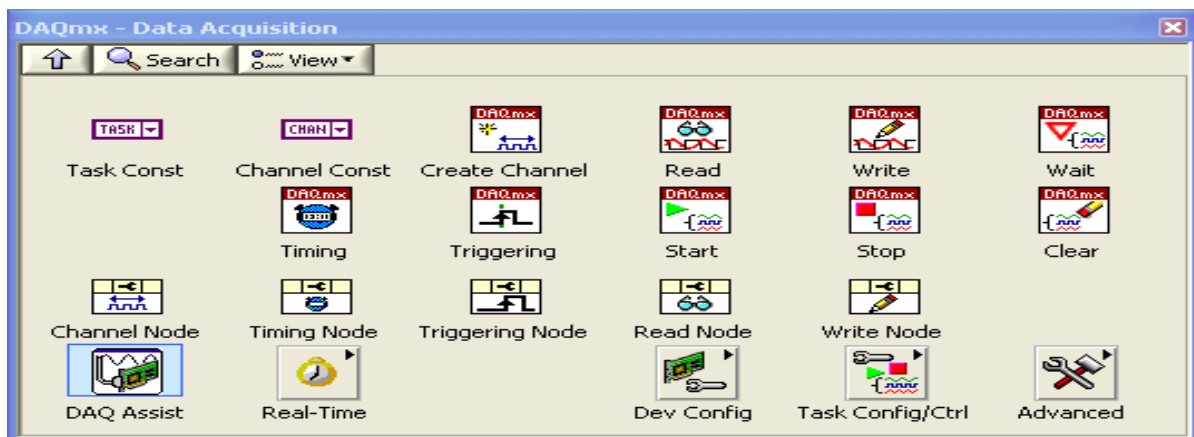


Figura 4. Selectarea NI DAQmx din tools-ul Function.

- DAQmx – driver (toolset) Data Acquisition:



Principalele funcții API sunt:

- Creare Canal Virtual (DAQmx Create Virtual Channel) – unde se alocă resurse (intrare/ieșire analogică, intrări/ieșiri digitale, numărătoare) și se configurează tipul de măsurători (de la senzori de accelerație, termocupluri, punți tensiometrice, măsurare

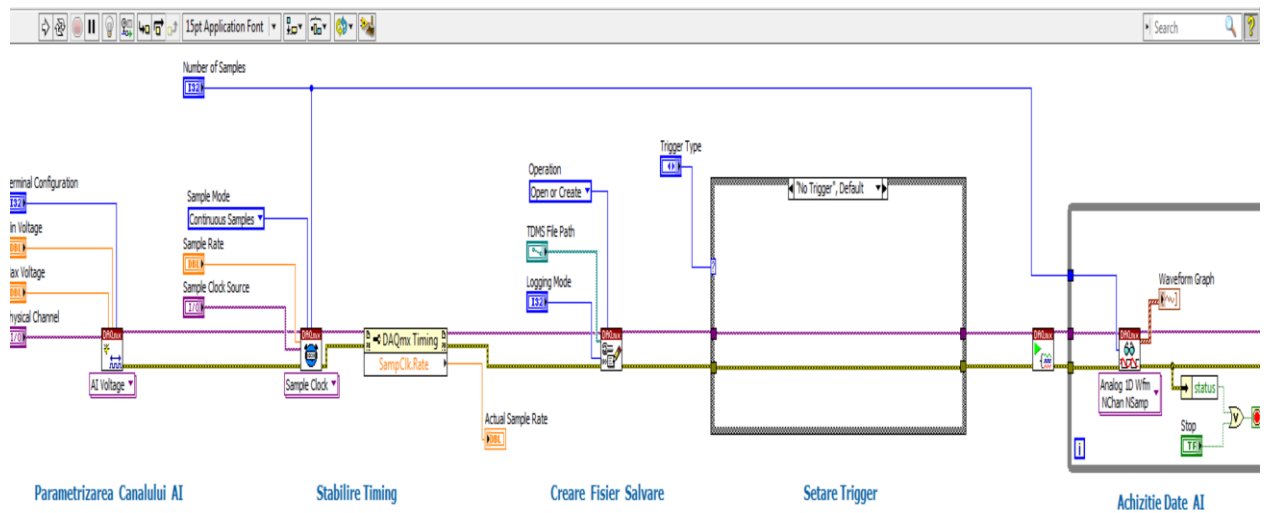


de frecvență, numărare de impulsuri, calculare poziție de le encodere în cuadratură, etc.);

- Configurarea parametrilor de temporizare a achiziției;
- Configurarea de Trigger;
- Citire/Scriere date în funcție de modul configurat (achiziție sau generare);
- Funcții de control a achiziției (Start, Stop, Dezactivarea resurse).

## Realizarea interfetelor de programare pentru achiziție (AI 1 canal) / generare (AO 2 canale simultane)

Structura software pentru achiziția de semnal (AI) este prezentată în figura 5.



**Figura 5.** Structura software-ului pentru achiziția de semnal.

Structura software-ului pentru generarea semnalelor sinusoidale continue este prezentată în figura 6.

## Interfața Grafică

În figura 7 este prezentată interfața grafică pentru aplicația de generare – achiziție semnale utilizând hardware DAQ USB 6351.

Etapele de configurare / executare aplicație

1. Selecarea canalelor fizice corespunzătoare plăcii de achiziție DAQ USB 6351. Se configurează task-ul de ieșire AO0:1 și task-ul de intrare AI0.

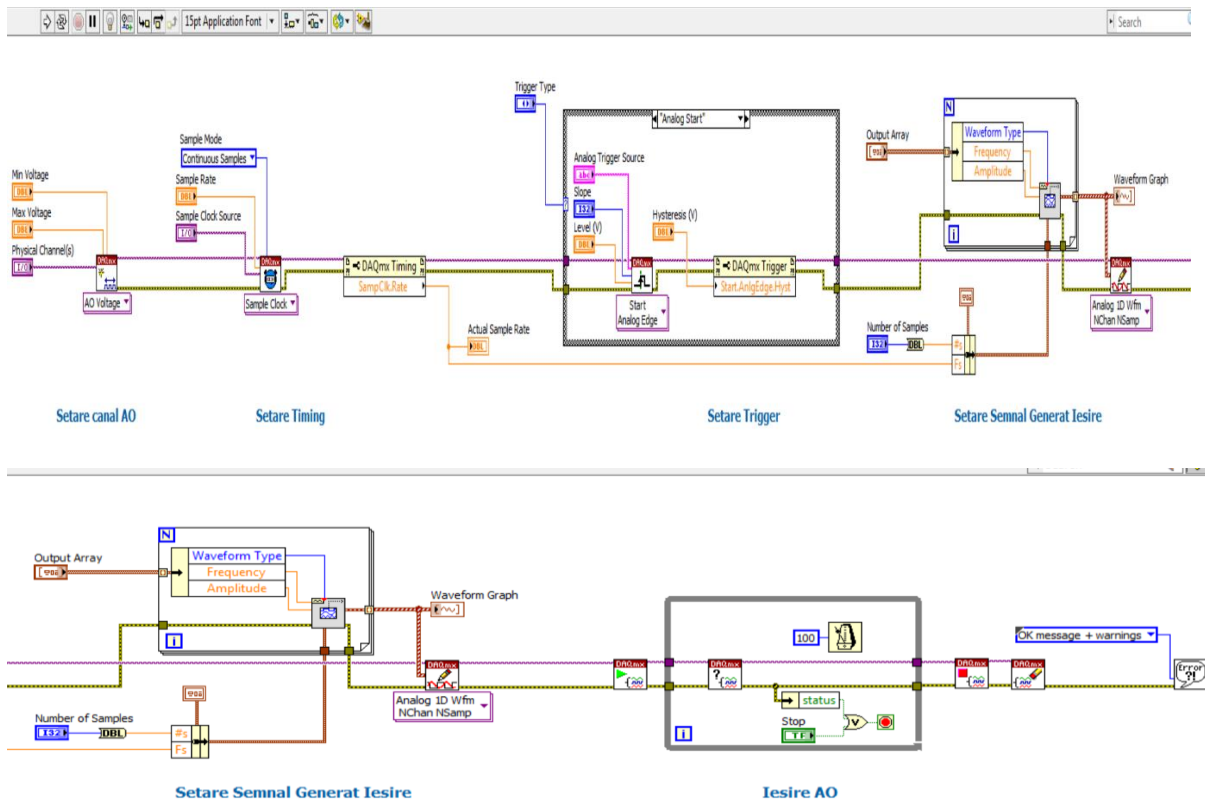


Figura 6. Structura software pentru generarea de semnale sinusoidale.

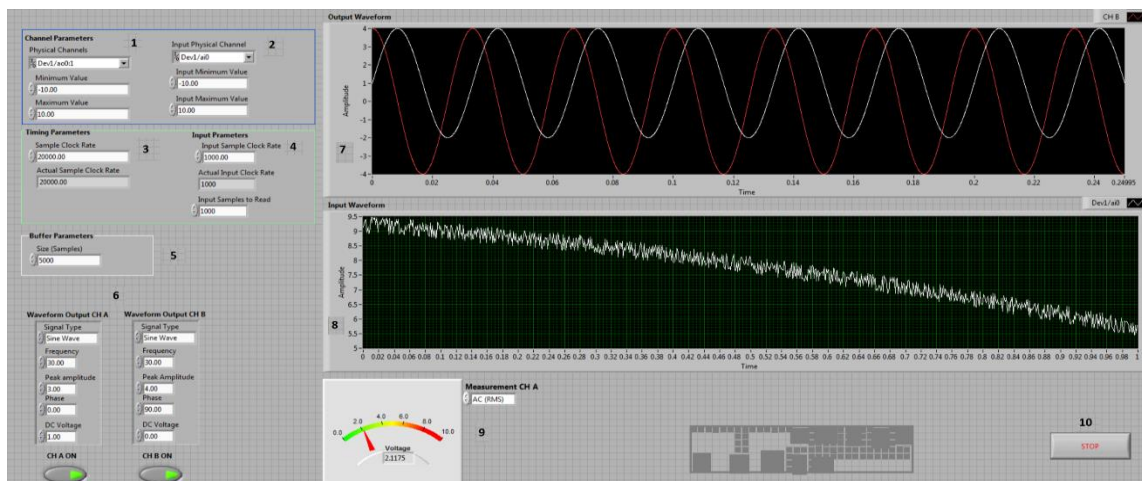


Figura 7. Interfața grafică pentru aplicația generare / achiziție semnale.

2. Introducerea intervalului minim și maxim de tensiune aferente canalelor fizice de intrare / ieșire. Conform prezentării tehnice valorile de configurare sunt:  $\pm 10 \text{ V}$  și  $\pm 5 \text{ V}$ .
3. Introducerea frecvenței de eșantionare (Sample Clock Rate) funcție de frecvențele aferente semnalelor sinusoidale care se generează (frecvența de eșantionare maximă pentru 2 canale de generare este de  $2 \text{ MS/s}$ ). Valori mai mari ale frecvenței de eșantionare va produce forme de semnal mai line.

4. Introducerea frecvenței de eșantionare (Sample Clock Rate) funcție de frecvențele aferente semnalelor de achiziție.
5. Introduceți dimensiunea buffer-ului de ieșiri analogice. Un tampon mai mare al buffer-ului ajută operațiile de bază de tip non-generare și evită erorile datorate vitezei procesorului. Dimensiunea buffer-ului de ieșire trebuie să fie minim egală cu cantitatea de date care sunt generate la fiecare iterație. Dimensiunea recomandată a buffer-ului este de cel puțin 2 ori mai mare decât cantitatea de date care sunt generate la fiecare iterație.
6. Introducerea informațiilor despre parametrii care definesc semnalele sinusoidale generate. Parametrii configurabili pentru semnalele sinusoidale de ieșire, sunt: frecvența [Hz], amplitudinea [ $V_{vv}$ ], faza [grade] și nivel DC [V].
7. Osciloscop pentru vizualizarea celor două sinusoidale generate.
8. Osciloscop pentru vizualizarea semnalului achiziționat.

Buton STOP pentru realizarea funcției OFF aplicație.

### **C. Concluzii**

În cadrul prezentei etape au fost realizate activități privind dezvoltarea în fază primară a modului electronic de control a capului de citire al instrumentului HBPSKP (Partea I) și au fost finalizate operațiile pentru dezvoltarea aplicației software de control și analiză de date a capului de măsură al instrumentului HBPSKP (Partea II).

Arhitectura modului electronic front-end (**FEE**) implică utilizarea următoarelor submodule: (Sub)Modulul SG (Signal Generation) al modului SGSAQ (Signal Generation and Signal Acquisition), (sub)modulul CDS (Coil Driving Signal), (sub)modulul TDS (Tip Driving Signal), (sub)modulul TSR (Tip Signal Readout), (sub)modulul SAQ (Signal Acquisition) al modului SGSAQ.

Aplicația software de control și analiză de date a capului de măsură al instrumentului HBPSKP a fost elaborată în două variante: una, dezvoltată în limbajul de programare C/C++ sub mediul integrat Microsoft Visual Studio Community 2015 și o a doua sub mediul integrat LabView. În cadrul ambelor variante sunt utilizate două task-uri. În cadrul primului task sunt aplicate două semnale de ieșire pe circuitele de output ale plăcii de achiziție National Instruments NI USB – 6351. În cadrul celui de-al doilea task, semnalul electric furnizat de un generator de semnal cu două canale este achiziționat prin intermediul aceleiași plăci de achiziție NI USB – 6351.

**Obiectivele prezentei etape au fost realizate integral.**

## D. Bibliografie

- [ 1] <https://www.visualstudio.com/>
- [ 2] <https://www.visualstudio.com/downloads/>
- [ 3] <http://zone.ni.com/reference/en-XX/help/370471AE-01/daqmxcfunc/daqmxcreatetask/>
- [ 4] <http://zone.ni.com/reference/en-XX/help/370471AE-01/daqmxcfunc/daqmxcreateaovoltagechan/>
- [ 5] <http://zone.ni.com/reference/en-XX/help/370471AE-01/daqmxcfunc/daqmxcfgsampclktiming/>
- [ 6] <http://zone.ni.com/reference/en-XX/help/370471AE-01/daqmxcfunc/daqmxwriteanalogf64/>
- [ 7] <http://zone.ni.com/reference/en-XX/help/370471W-01/daqmxcfunc/daqmxstarttask/>
- [ 8] <http://zone.ni.com/reference/en-XX/help/370471AE-01/daqmxcfunc/daqmxreadanalogf64/>
- [ 9] <http://zone.ni.com/reference/en-XX/help/370471AE-01/daqmxcfunc/daqmxstoptask/>
- [10] <http://zone.ni.com/reference/en-XX/help/370471AE-01/daqmxcfunc/daqmxcleartask/>
- [11] <http://romania.ni.com/labview>