

# [Π3.2] Λογισμικό Διασύνδεσης Συσκευών

Επίπεδο Διάχυσης:	Δημόσιο
Συμφωνηθείσα Ημερομηνία:	Μήνας 11, 27/06/2021
Πραγματική Ημερομηνία:	Μήνας 12, 29/07/2021
Ενότητα Εργασίας:	ΕΕ3 Λογισμικό Διασύνδεσης Συσκευών
Υποενότητα:	Ε3.2 Υλοποίηση Συστήματος και Βιβλιοθήκη Λογισμικού
Τύπος:	Λογισμικό
Στάδιο Έγκρισης:	τελικό
Έκδοση:	v1.0
Πλήθος Σελίδων:	26
Ονομασία Αρχείου:	D3_2_v1.0_final.docx

#### Περίληψη:

Το παρόν έγγραφο αποτελεί την τεχνική τεκμηρίωση του λογισμικού που υποστηρίζει την διασύνδεση των αισθητήρων. Ιδιαίτερη έμφαση δίνεται στην επικοινωνία των αισθητήρων με το ενδιάμεσο λογισμικό (middleware) του NeuroMkt που επιτυγχάνει τον συγχρονισμό των ετερογενών σημάτων. Σκοπός του παρόντος εγγράφου αποτελεί η παροχή οδηγιών που σχετίζονται με την μεταγλώττιση, εγκατάσταση και χρήση του λογισμικού που υποστηρίζει την διασύνδεση των αισθητήρων και γενικότερα όλων των συσκευών που εμπλέκονται στις καταγραφές του NeuroMkt. Πέρα από τις παραπάνω οδηγίες, παρέχονται πληροφορίες σχετικά με την οργάνωση του αποθετηρίου, συνήθη προβλήματα και την επίλυση τους, ενώ ταυτόχρονα παρατίθενται και μερικά παραδείγματα χρήσης.

Οι πληροφορίες που περιέχονται σε αυτό το έγγραφο αντικατοπτρίζουν μόνο τις απόψεις του δημιουργού και η Ευρωπαϊκή Ένωση/ΕΥΔΕ-ΕΤΑΚ δεν ευθύνεται για τυχόν χρήση των πληροφοριών που περιέχονται σε αυτήν. Οι πληροφορίες στο παρόν έγγραφο παρέχονται ως έχουν και δεν παρέχεται καμία εγγύηση ότι οι πληροφορίες είναι κατάλληλες για οποιοδήποτε συγκεκριμένο σκοπό. Ο χρήστης χρησιμοποιεί αυτές τις πληροφορίες με αποκλειστικό κίνδυνο και ευθύνη.

# Πνευματική ιδιοκτησία

© Πνευματική ιδιοκτησία 2020 της κοινοπραξίας NeuroMkt που αποτελείται από:

1. Εθνικό Κέντρο Έρευνας και Τεχνολογικής Ανάπτυξης (ΕΚΕΤΑ)
2. Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης (ΑΠΘ)
3. MMS ΔΙΑΦΗΜΙΣΤΙΚΗ Α.Ε (MMS)
4. ΔΙΑΜΑΝΤΗΣ ΜΑΣΟΥΤΗΣ Α.Ε. ΣΟΥΠΕΡ ΜΑΡΚΕΤ (ΜΑΣΟΥΤΗΣ)

ΑΠΑΓΟΡΕΥΕΤΑΙ αυστηρά η αναδημοσίευση, ολική, μερική ή περιληπτική αναπαραγωγή, η κατά παράφραση ή διασκευή των κειμένων που περιέχονται στο παρόν έγγραφο με οποιονδήποτε τρόπο (ηλεκτρονικό, μηχανικό, φωτοτυπικό, ηχογράφησης ή άλλο), χωρίς προηγούμενη γραπτή άδεια των μελών της κοινοπραξίας.

Με επιφύλαξη παντός δικαιώματος.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Ταμείο  
Περιφερειακής Ανάπτυξης



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΥΠΟΥΡΓΕΙΟ  
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ  
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΙΧ & ΤΣ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΙΧ



ΕΣΠΑ  
2014-2020  
ανάπτυξη • εργασία • αλληλεγγύη

Υλοποιήθηκε στο πλαίσιο της Δράσης ΕΡΕΥΝΩ-ΔΗΜΙΟΥΡΓΩ-ΚΑΙΝΟΤΟΜΩ και συγχρηματοδοτήθηκε από την Ευρωπαϊκή Ένωση και εθνικούς πόρους μέσω του Ε.Π. Ανταγωνιστικότητα, Επιχειρηματικότητα & Καινοτομία (ΕΠΑνΕΚ) (κωδικός έργου: Τ2ΕΔΚ-03661)

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

## ΙΣΤΟΡΙΚΟ ΕΓΓΡΑΦΟΥ

Έκδοση	Ημερομηνία	Λόγος Αναθεώρησης	Αναθεωρήθηκε από
v0.1 (alpha)	27/05/2021	Αρχική έκδοση με τον Πίνακα Περιεχομένων για να διαμοιραστεί στην κοινοπραξία και να εγκριθεί από τον συντονιστή	Φώτης Καλαγάνης
v0.2 (beta)	16/07/2021	Ολοκλήρωση προσθήκης περιεχομένου και αποστολή προς εσωτερική αξιολόγηση	Φώτης Καλαγάνης
v1.0 (final)	20/07/2021	Διορθώσεις εσωτερικού αξιολογητή	Κυριάκος Ρίσκος
v1.0 (final)	26/07/2021	Αναθεώρηση/Επικύρωση	Σπύρος Νικολόπουλος
v1.0 (final)	29/07/2021	Έγκριση από τον επιστημονικό υπεύθυνο	Ιωάννης Κομπατσιάρης

## ΛΙΣΤΑ ΣΥΝΕΙΣΦΕΡΟΝΤΩΝ

Φορέας	Όνομα	Στοιχεία Επικοινωνίας
ΕΚΕΤΑ	Φώτης Καλαγάνης	<a href="mailto:fkalaganis@iti.gr">fkalaganis@iti.gr</a>
ΕΚΕΤΑ	Κώστας Γεωργιάδης	<a href="mailto:kostas.georgiadis@iti.gr">kostas.georgiadis@iti.gr</a>
ΑΠΘ	Κυριάκος Ρίσκος	<a href="mailto:riskoskyriakos@gmail.com">riskoskyriakos@gmail.com</a>
ΕΚΕΤΑ	Σπύρος Νικολόπουλος	<a href="mailto:nikolopo@iti.gr">nikolopo@iti.gr</a>
ΕΚΕΤΑ	Ιωάννης Κομπατσιάρης	<a href="mailto:ikom@iti.gr">ikom@iti.gr</a>

## ΚΥΡΙΟΤΕΡΑ ΣΗΜΕΙΑ

Το συγκεκριμένο έγγραφο συνοδεύει την πρωταρχική έκδοση του λογισμικού που υποστηρίζει την διασύνδεση των συσκευών/αισθητήρων.

Οι βασικοί άξονες του παραδοτέου σχετίζονται με τις οδηγίες μεταγλώττισης και τις οδηγίες εγκατάστασης. Αρχικά, περιγράφεται αναλυτικά η διαδικασία που απαιτείται για τη μεταγλώττιση του πηγαίου κώδικα σε εκτελέσιμα αρχεία. Κατόπιν, δίνεται αναλυτική περιγραφή για τη χρήση των παραχθέντων εκτελέσιμων αρχείων.

Πέρα από τα ανωτέρω, προκειμένου να γίνει εμφανής η χρησιμότητα του λογισμικού διασύνδεσης των συσκευών δίνονται μερικά παραδείγματα χρήσης τους.

Καθώς το λογισμικό του NeuroMkt είναι διαθέσιμο, επιγράφεται η δομή και η οργάνωση του αποθετηρίου που φιλοξενεί τον πηγαίο κώδικα του έργου.

Τέλος, παρουσιάζονται μερικές από τις πιο συνήθεις ερωτήσεις και κοινότυπα προβλήματα που ενδεχομένως να παρουσιαστούν κατά τη μεταγλώττιση ή χρήση του λογισμικού διασύνδεσης των συσκευών.

## ΣΥΝΤΜΗΣΕΙΣ ΚΑΙ ΑΚΡΩΝΥΜΙΑ

<b>API</b>	Application Programming Interface
<b>ET</b>	Eye Tracker
<b>EEG</b>	Electroencephalography
<b>LSL</b>	Lab Streaming Layer
<b>SDK</b>	Software Development Kit

## ΠΕΡΙΕΧΟΜΕΝΑ

1	Εισαγωγή	7
2	Ανασκόπηση του συστήματος NeuroMkt	8
2.1	Αισθητήρες Συστήματος	9
2.2	Middleware	10
3	Μεταγλώττιση Προγραμμάτων	11
3.1	Διασύνδεση Συσκευών Καταγραφής	11
3.1.1	Περιβάλλον Εργασίας	11
3.1.2	Εξαρτήσεις	11
3.1.3	Ρυθμίσεις Περιβάλλοντος	12
3.1.4	Οδηγίες Μεταγλώττισης	12
3.2	SDK Ανάλυσης Σημάτων και DashBoard	14
4	Εγκατάσταση	15
4.1	Προδιαγραφές	15
4.2	Πακέτα	16
4.3	Οδηγίες Εγκατάστασης και Διασύνδεσης	19
4.4	Οργάνωση Και Χρήση Κώδικα στο GitHub	21
5	Παραδείγματα Χρήσης	22
5.1	Live Viewer	22
5.2	DashBoard	22
6	Επίλυση Προβλημάτων και Συχνές Ερωτήσεις	24
6.1	Προσθήκη νέου Αισθητήρα	24

## 1 ΕΙΣΑΓΩΓΗ

Το παρόν έγγραφο περιγράφει την τεκμηρίωση για την πρωταρχική έκδοση του λογισμικού που υποστηρίζει την διασύνδεση συσκευών του NeuroMkt (ο κώδικας είναι διαθέσιμος στο αποθετήριο του έργου<sup>1</sup>). Το λογισμικό διασύνδεσης βασίζεται στην C++ και χρησιμοποιεί το Qt framework προκειμένου να παρέχει ένα γραφικό περιβάλλον. Το λογισμικό αυτό επεκτείνεται στο περιβάλλον του Matlab μέσω της κατάλληλα διαμορφωμένης έκδοσης της liblsl για το Matlab.

Η δομή του εγγράφου έχει ως εξής: Στο Κεφάλαιο 2 γίνεται μια ανασκόπηση της αρχιτεκτονικής του συστήματος όπως παρουσιάστηκε στο παραδοτέο Π3.1. Καθώς το παρόν έγγραφο αφορά το λογισμικό που υποστηρίζει την διασύνδεση των συσκευών, η αναφορά στην αρχιτεκτονική περιορίζεται στους αισθητήρες και στο middleware. Στο Κεφάλαιο 3 περιγράφεται αναλυτικά η διαδικασία που απαιτείται για την μεταγλώττιση των προγραμμάτων που υποστηρίζουν την διασύνδεση των αισθητήρων. Στο Κεφάλαιο 4 παρουσιάζεται η διαδικασία εγκατάστασης των παραπάνω προγραμμάτων. Στο Κεφάλαιο 5 παρατίθενται μερικά παραδείγματα χρήσης ώστε να γίνει εμφανής η χρήση του λογισμικού διασύνδεσης συσκευών. Τέλος, στο Κεφάλαιο 6, γίνεται η περιγραφή του αποθετηρίου που φιλοξενεί τον κώδικα του έργου και παρατίθενται μερικές συχνές ερωτήσεις που συνοδεύονται από τις αντίστοιχες απαντήσεις.

---

<sup>1</sup> <https://github.com/NeuroMkt/Middleware>



## 2 ΑΝΑΣΚΟΠΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ NEUROMKT

Τα λογικά στοιχεία που συνθέτουν την αρχιτεκτονική του NeuroMkt είναι τα Layers, τα Modules και τα Interfaces. Τα επίπεδα χρησιμοποιούνται για να υποδηλώσουν τα μέρη του συστήματος που εξυπηρετούν διαφορετικό σκοπό. Τα modules χρησιμοποιούνται για να περιγράψουν τις βασικές λειτουργίες που εκτελούνται σε κάθε επίπεδο και τα interfaces χρησιμοποιούνται για να καθορίσουν τον τρόπο ροής των πληροφοριών από το ένα στρώμα στο άλλο. Στην Εικόνα 1 παρουσιάζεται μια αφαιρετική απεικόνιση της αρχιτεκτονικής του συστήματος όπως αυτή αναλύεται στο Π3.1.



Εικόνα 1. Τα διακριτά layers της αρχιτεκτονικής του λογισμικού του NeuroMkt.

Το πρώτο layer είναι το layer αισθητήρων που εξυπηρετεί το ρόλο της καταγραφής του σήματος από τις συσκευές αισθητήρων για να το ωθήσει στην αρχιτεκτονική μας. Εκτός από το υλικό μέρος (EEG recorder, Eye-tracker και άλλα), μεταξύ των modules του layer αυτού ταξινομούμε επίσης τα προγράμματα οδήγησης (ένα για κάθε συσκευή υλικού) που είναι απαραίτητα για τη διάθεση των παραγόμενων σημάτων για το middleware του NeuroMkt. Αυτά τα προγράμματα οδήγησης έχουν κατασκευαστεί χρησιμοποιώντας το SDK κάθε συσκευής έτσι ώστε να παρέχουν τα σήματα σε μια δομή και μορφή κατάλληλη για το middleware του NeuroMkt.

Το δεύτερο επίπεδο είναι το middleware. Ο ρόλος αυτού του επιπέδου είναι να λειτουργεί ως μεσολαβητής μεταξύ των συσκευών αισθητήρων και της υπόλοιπης αρχιτεκτονικής του NeuroMkt. Με άλλα λόγια, εφ' όσον μια νέα συσκευή αισθητήρα πληροί τις προδιαγραφές εισόδου του ενδιάμεσου λογισμικού, το υπόλοιπο σύστημα θα πρέπει να λειτουργεί απρόσκοπτα. Στην αρχιτεκτονική του NeuroMkt, το λογισμικό LabStreamingLayer έχει επιλεγεί για να χρησιμεύσει ως ενδιάμεσο λογισμικό NeuroMkt. Το middleware αποτελείται από τρία modules (μονάδες): α) την Απόκτηση Σήματος, β) την Χρονική Επισημείωση και γ) τον Συγχρονισμό. Η Απόκτηση Σήματος είναι η μονάδα που είναι υπεύθυνη για τη λήψη σημάτων από το επίπεδο αισθητήρων. Ο ρόλος της Χρονικής Επισημείωσης είναι η προσθήκη χρονικών δεικτοδοτήσεων στα ληφθέντα σήματα. Τέλος, το πιο σημαντικό module

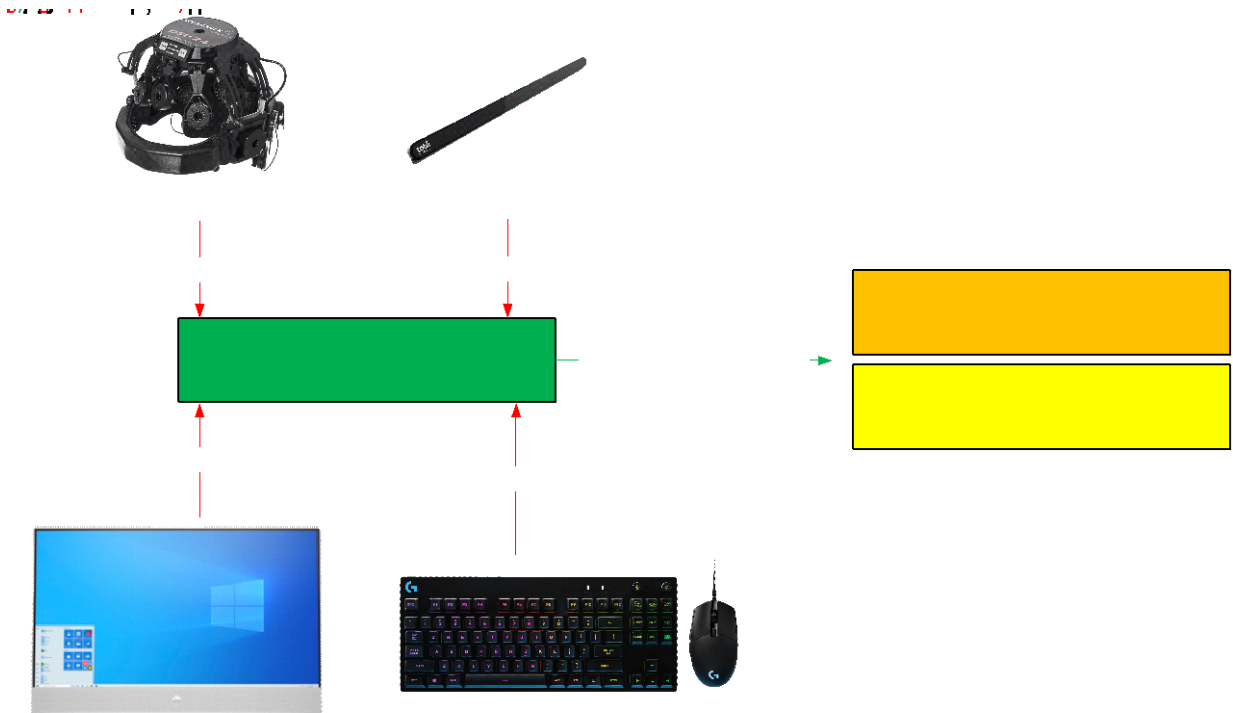
του middleware είναι το module Συγχρονισμού που φροντίζει για το συγχρονισμό των (αρχικά αποσυγχρονισμένων) σημάτων με βάση τις χρονικές τους δεικτοδοτήσεις. Ως αποτέλεσμα, τα σήματα μπορούν να προωθηθούν περαιτέρω στην αρχιτεκτονική του NeuroMkt σε συγχρονισμένη λειτουργία.

Το τρίτο επίπεδο στην αρχιτεκτονική μας είναι το SDK Ανάλυσης Σημάτων. Ο ρόλος αυτού του επιπέδου είναι τριπλός: α) επικοινωνεί με διαφάνεια με το middleware, έτσι ώστε να λαμβάνει τα συγχρονισμένα σήματα, β) εφαρμόζει μια εκτενή λίστα διαδικασιών και μεθόδων για τη μετάφραση των φυσιολογικών αποκρίσεων σε κατάλληλους βιοδείκτες, και γ) επικοινωνεί αυτούς τους βιοδείκτες στο dashboard.

Το τελευταίο επίπεδο της αρχιτεκτονικής του NeuroMkt είναι το Dashboard το οποίο απεικονίζει με έναν εύληπτο τρόπο όλους εκείνους τους βιοδείκτες που είναι ικανοί να υποβοηθήσουν τους χρήστες-διαφημιστές στην αξιολόγηση των διαφημιστικών μηνυμάτων. Σημειώνεται ότι, αν και στην παρούσα φάση του έργου δεν κρίνεται αναγκαίο, έχει υπάρξει πρόβλεψη για αμφίδρομη επικοινωνία του Dashboard με όλα τα υπόλοιπα επίπεδα της εφαρμογής μέσω sockets.

## 2.1 ΑΙΣΘΗΤΗΡΕΣ ΣΥΣΤΗΜΑΤΟΣ

Τα SDK των αισθητήρων, είναι το μέσο επικοινωνίας ανάμεσα στον υπολογιστή και τον αισθητήρα και αποτελεί τη βάση για να στηθεί η αρχιτεκτονική του συστήματος. Πρόκειται ουσιαστικά για λογισμικό που παρέχεται από τον κατασκευαστή του εκάστοτε αισθητήρα και επιτρέπει την συλλογή και διάθεση των δεδομένων. Ταυτόχρονα, τα SDK αισθητήρων ενσωματώνουν ένα σύνολο λειτουργιών που είναι απαραίτητα ώστε να μετατρέψουν τα δεδομένα που καταγράφουν οι αισθητήρες σε ποιοτικότερα και (ενδεχομένως) σημασιολογικά ανώτερα σήματα. Τέλος, τονίζουμε ότι το SDK είναι το λογισμικό που επικοινωνεί ευθέως με το firmware του αισθητήρα και ορίζει την διαθεσιμότητα των δεδομένων (π.χ. ποια δεδομένα παρέχονται στον χρήστη-προγραμματιστή και ποια αποκρύπτονται) και την αμφίδρομη επικοινωνία (αλλαγή συχνότητας δειγματοληψίας, αλλαγή ψηφιακής ανάλυσης, κ.α.). Στις δύο επόμενες ενότητες αναλύουμε τις διαδικασίες συλλογής δεδομένων για ET και EEG και παρουσιάζουμε ενδεικτικές λειτουργίες των SDK που συνήθως συνοδεύουν αυτούς τους αισθητήρες.



Εικόνα 2. Διάγραμμα ροής πληροφορίας από τους αισθητήρες στο ενδιάμεσο λογισμικό και στη συνέχεια στα ανώτερα επίπεδα τις αρχιτεκτονικής.

## 2.2 MIDDLEWARE

Το δεύτερο layer αποτελεί το middleware του NeuroMkt. Σκοπός του δεύτερου αυτού επιπέδου είναι να αποτελέσει τον μεσολαβητή ανάμεσα στις συσκευές των αισθητήρων και των υπόλοιπων μερών της αρχιτεκτονικής του NeuroMkt. Πιο συγκεκριμένα, εάν ένας καινούριος αισθητήρας πληροί τις προδιαγραφές εισόδου (σε ότι αφορά το ενδιάμεσο λογισμικό), το υπόλοιπα μέρη του συστήματος θα πρέπει να είναι σε θέση να λειτουργήσουν απρόσκοπτα. Για αυτόν το λόγο, το λογισμικό LabStreamingLayer έχει επιλεγεί για να αποτελέσει τη βάση για το ενδιάμεσο λογισμικό του NeuroMkt. Το middleware απαρτίζεται από τρία διακριτά modules: α) την Απόκτηση Σήματος, β) την Χρονική Επισημείωση και γ) τον Συγχρονισμό. Η πρώτη μονάδα είναι υπεύθυνη για τη λήψη σημάτων από το επίπεδο αισθητήρων. Η δεύτερη μονάδα αφορά την προσθήκη χρονικών δεικτοδοτήσεων στα ληφθέντα σήματα. Τέλος, η πιο σημαντική μονάδα είναι η ομάδα συγχρονισμού, που επιτελεί το συγχρονισμό των (αρχικά αποσυγχρονισμένων) σημάτων σύμφωνα με τις χρονικές τους δεικτοδοτήσεις. Συνεπώς, αφού τα σήματα περάσουν μέσα από μέσα από το ενδιάμεσο λογισμικό μπορούν να προωθηθούν στα περαιτέρω μέρη της αρχιτεκτονικής του NeuroMkt.

### 3 ΜΕΤΑΓΛΩΤΤΙΣΗ ΠΡΟΓΡΑΜΜΑΤΩΝ

Η παρούσα ενότητα περιέχει όλες τις απαραίτητες οδηγίες προκειμένου να μπορέσει κάποιος να μεταγλωττίσει, από τον πηγαίο κώδικα, τα προγράμματα που υποστηρίζουν τη διασύνδεση συσκευών.

#### 3.1 ΔΙΑΣΥΝΔΕΣΗ ΣΥΣΚΕΥΩΝ ΚΑΤΑΓΡΑΦΗΣ

##### 3.1.1 ΠΕΡΙΒΑΛΛΟΝ ΕΡΓΑΣΙΑΣ

Η libsl καθώς επίσης και οι περισσότερες εφαρμογές χρησιμοποιούν το CMAKE και την C++. Οι περισσότερες εφαρμογές προκειμένου να παρέχουν ένα γραφικό περιβάλλον εργασίας χρησιμοποιούν την Qt. Για την μεταγλώττιση των προγραμμάτων συστήνεται η εγκατάσταση της πιο πρόσφατης έκδοσης των εργαλείων. Ο παρακάτω πίνακας περιέχει όλες τις πλατφόρμες που υποστηρίζει η libsl και συνεπώς μπορεί να μεταγλωττιστεί.

Πίνακας 1. Πλατφόρμες που υποστηρίζει η libsl.

Σύστημα/Μεταγλωττιστής	Υποστηριζόμενη έκδοση	
Όνομα	Min	Max
Windows	Xp	10 (20H2)
Visual C++	2015	2019 (16.6)
OS X	10.9	10.14
XCode	9.3	
Ubuntu	16.04	20.10
CentOS	6	
Clang	3.5	9.0.0
g++	6.2	9.0
Alpine Linux	<3.6	3.12
CMake	3.12	3.18

Όπως φαίνεται και στον παραπάνω πίνακα η libsl δουλεύει ακόμα και σε πολύ παλιά (π.χ. Windows XP) ή πολύ μικρά (π.χ. Raspberry Pi, Android, κτλ.) συστήματα. Ωστόσο, ορισμένες εφαρμογές μπορεί να έχουν μεγαλύτερες απαιτήσεις.

##### 3.1.2 ΕΞΑΡΤΗΣΕΙΣ

Οι εφαρμογές που επιτρέπουν τη διασύνδεση των συσκευών εξαρτώνται από third party βιβλιοθήκες. Προκειμένου κάποιος να μεταγλωττίσει τις εφαρμογές αυτές, απαιτείται η εγκατάσταση των παρακάτω βιβλιοθηκών:

- **Qt:** Οι προτεινόμενες εκδόσεις για την μεταγλώττιση είναι οι Qt5 ή Qt6. Η Qt χρειάζεται μόνο για τις εφαρμογές που έχουν γραφικό περιβάλλον. Προκειμένου να γίνει η μεταγλώττιση των εφαρμογών με την Qt, απαιτείται η εγκατάσταση. Σε περίπτωση που το CMake δεν μπορέσει αυτόματα να εντοπίσει την τοποθεσία εγκατάστασης χρειάζεται είτε ο χρήστης να προσθέσει στο system path το αρχείο για τον συγκεκριμένο μεταγλωττιστή, `set PATH=C:Qt<version><compiler_arch>;%PATH%` ή να προσθέσει την τοποθεσία εγκατάστασης της βιβλιοθήκης Qt μέσα στις ρυθμίσεις του CMake εκτελώντας την εντολή `DQt5_DIR=C:/Qt/<version>/<compiler_arch>/lib/cmake/Qt5/` για την Qt5 και την εντολή `DQt6_DIR=C:/Qt/<version>/<compiler_arch>/lib/cmake/Qt6/` για την Qt6.

`DQt6_DIR=C:/Qt/<version>/<compiler_arch>/lib/cmake/Qt6/` για την Qt6. Η Qt σε Windows μπορεί να εγκατασταθεί μέσω του επίσημου προγράμματος εγκατάστασης<sup>2</sup>.

- **Boost:** Η Boost χρησιμοποιείται κατά κύριο λόγο προκειμένου να συνδέσει συσκευές στο τοπικό δίκτυο μέσω της Boost.Asio. Για την μεταγλώττιση συνήθως αρκεί να ο χρήστης να κατεβάσει την Boost<sup>3</sup>, να την αποσυμπιέσει και να υποδείξει το μονοπάτι στο CMake εκτελώντας `-DBOOST_ROOT=path/to/boost`

### 3.1.3 ΡΥΘΜΙΣΕΙΣ ΠΕΡΙΒΑΛΛΟΝΤΟΣ

Σε περιβάλλον Windows, προκειμένου να μεταγλωττιστούν οι εφαρμογές που υποστηρίζουν τη διασύνδεση συσκευών, το παρακάτω block εντολών χρειάζεται να οριστεί ως ρύθμιση στο πρόγραμμα εγκατάστασης του Visual Studio 2019.

```
{
  "version": "1.0",
  "components": [
    "microsoft.visualstudio.component.vc.coreide",
    "microsoft.visualstudio.component.vc.cmake.project"
  ]
}
```

Οι εκδόσεις από Visual Studio 2017 και έπειτα συμπεριλαμβάνουν το CMake. Σε περίπτωση που κάποιος όμως επιθυμεί μια ξεχωριστή εγκατάσταση μπορεί να κατεβάσει το ανεξάρτητο πρόγραμμα εγκατάστασης<sup>4</sup>.

Προτείνεται κατά την διάρκεια εγκατάστασης να επιλεγεί το `Add to the path (for all users)`.

### 3.1.4 ΟΔΗΓΙΕΣ ΜΕΤΑΓΛΩΤΤΙΣΗΣ

Αφού έχουμε στήσει το περιβάλλον εργασίας, οι παρακάτω οδηγίες περιγράφουν λεπτομερώς τη διαδικασία μεταγλώττισης των εφαρμογών για την διασύνδεση των συσκευών. Οι παρακάτω πληροφορίες αν και απευθύνονται κατά κύριο λόγο στις εφαρμογές που χρησιμοποιεί το NeuroMkt, μπορούν στο ακέραιο να χρησιμοποιηθούν ώστε να μεταγλωττιστεί η μεγάλη πλειοψηφία των εφαρμογών που περιέχονται στο αποθετήριο του Lab Streaming Layer.

1. Βεβαιωθείτε πως το περιβάλλον εργασίας έχει στηθεί όπως περιγράφεται στις δύο παραπάνω ενότητες.
2. Δημιουργήστε τον πηγαίο κώδικα της εφαρμογής που θέλετε να μεταγλωττίσετε.
  - Αν δημιουργείτε μια καινούρια εφαρμογή με C++ για να υποστηρίξετε μια νέα συσκευή τότε πρέπει να κατεβάσετε το `template`<sup>5</sup>.
  - Σε περίπτωση εξωτερικής εφαρμογής, πρέπει να γίνει `git clone` το repository από το github.
3. Πρέπει να εκκαθαρίσετε το μονοπάτι που θα γίνει το `build`.
  - Μπορείτε να χρησιμοποιήσετε το γραφικό περιβάλλον για να σβήσετε τα περιεχόμενα του μονοπατιού ή αλλιώς την γραμμή εντολών όπως παρακάτω.
  - Ανοίξτε ένα `terminal/shell/command prompt` και μεταβείτε στο αντίστοιχο μονοπάτι.
  - Αν υπάρχει ήδη φάκελος διαγράψτε τον με την εντολή `rmdir /S build`
4. Ρυθμίστε το project χρησιμοποιώντας το περιβάλλον του LSL για το CMake.

<sup>2</sup> [http://download.qt.io/official\\_releases/online\\_installers/qt-unified-windows-x86-online.exe](http://download.qt.io/official_releases/online_installers/qt-unified-windows-x86-online.exe)

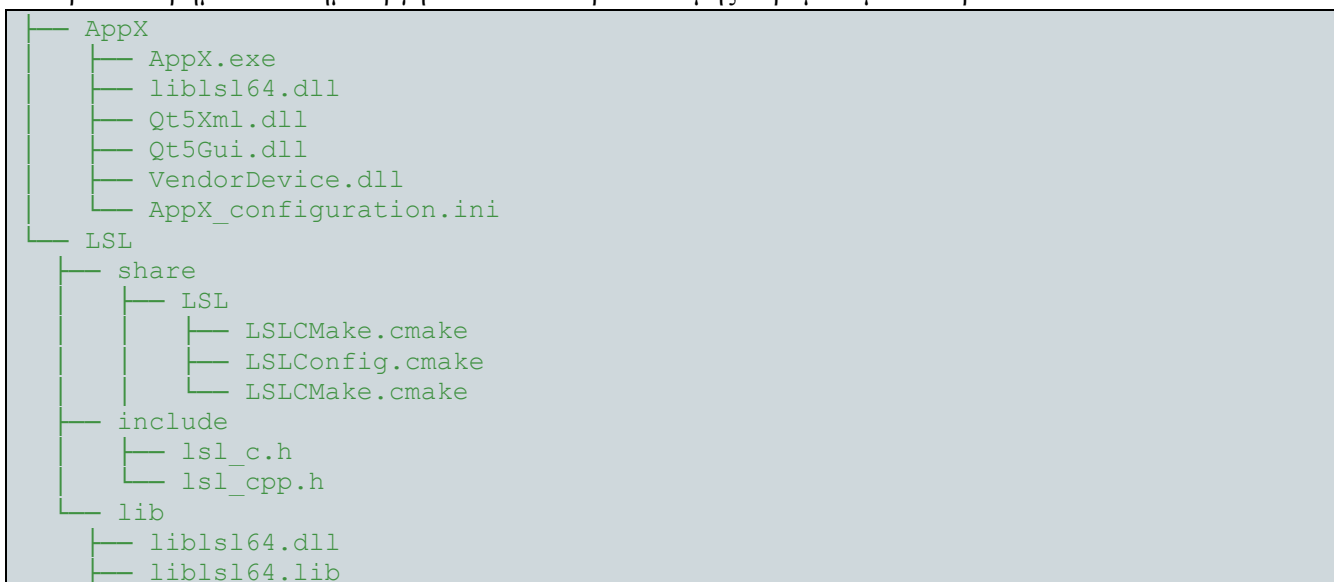
<sup>3</sup> <https://www.boost.org/users/download/>

<sup>4</sup> <https://cmake.org/download/>

<sup>5</sup> [https://github.com/labstreaminglayer/AppTemplate\\_cpp\\_qt/generate](https://github.com/labstreaminglayer/AppTemplate_cpp_qt/generate)

- Επιλογή 1: Προτεινόμενη Επιλογή για Windows - Visual Studio 2017 ή μεταγενέστερο λογισμικό.
  - Ανοίξτε τον φάκελο `CMakeLists.txt` στο Visual Studio `File->Open->CMake`
  - Αλλάξτε τις ρυθμίσεις για το CMake εκτελώντας τα ακόλουθα βήματα: `CMake/Project->Change CMake Settings`
  - Αλλάξτε το project από το αντίστοιχο αναδιπλούμενο μενού (`x64-Debug`, `x64-Release`). Αυτό θα πυροδοτήσει την επαναρρύθμιση του CMake με τις νέες μεταβλητές.
- Επιλογή 2: Μέσω της γραμμής εντολών.
  - Ανοίξτε ένα τερματικό
  - Εκτελέστε το CMake με τις παρακάτω παραμέτρους: `cmake -S . -B build -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL_PREFIX="build/install" other options`
- Επιλογή 3: Χρησιμοποιώντας το GUI του CMake.
  - Ανοίξτε ένα τερματικό (terminal/shell/command prompt) και αλλάξτε το μονοπάτι στον φάκελο που περιέχει το project.
  - Εκτελέστε την εντολή `cmake-gui -S . -B build`
  - Κάντε ένα αρχικό `Configure`. Σε περίπτωση που χρειάζεται να δημιουργηθεί ένας νέος φάκελος πατήστε αποδοχή.
  - Επιλέξτε μεταγλωττιστή και πατήστε `Finish`.
  - Χρησιμοποιείστε το γραφικό περιβάλλον για να προσθέσετε τις παρακάτω εξαρτήσεις πατώντας το `Add Entry`.
    - Qt5\_DIR ή Qt6\_DIR αν το μονοπάτι που προστέθηκε αυτόματα δεν είναι το σωστό.
    - Boost αν το προεγκατεστημένο μονοπάτι είναι λάθος.
    - Το μονοπάτι όπου τα διανεμημένα δυαδικά αρχεία είναι αντιγραμμένα `CMAKE_INSTALL_PREFIX`
    - Επιλέξτε τον τύπο του build (`Debug`, `Release`). Αυτή η επιλογή μπορεί να εφαρμοστεί και μέσα στο Visual Studio αργότερα.
  - Πατήστε το `Configure` ξανά προκειμένου να εφαρμοστούν οι αλλαγές.
  - Πατήστε το `Generate` για να δημιουργηθεί το Visual Studio Solution.

Τα παραπάνω βήματα θα δημιουργήσουν ένα δέντρο διανομής παρόμοιο με το παρακάτω:



## 3.2 SDK ΑΝΑΛΥΣΗΣ ΣΗΜΑΤΩΝ ΚΑΙ DASHBOARD

Τόσο το SDK ανάλυσης σημάτων όσο και το Dashboard έχουν υλοποιηθεί στο Matlab. Η μόνη εξάρτηση που υπάρχει αναφορικά με τη διασύνδεση των συσκευών είναι η liblsl. Στις περισσότερες των περιπτώσεων ο χρήστης μπορεί να κατεβάσει έτοιμη, pre-compiled, την liblsl για το Matlab. Παρόλα αυτά, οι διαθέσιμες pre-compiled εκδόσεις δεν καλύπτουν όλες τις υπάρχουσες διανομές του Matlab. Ως εκ τούτου, υπάρχει περίπτωση να χρειαστεί κάποιος να μεταγλωττίσει την liblsl για Matlab από την αρχή. Παρακάτω παρέχονται οι σαφείς οδηγίες για αυτή τη διαδικασία.

1. Κατεβάστε σαν αρχείο zip ή μέσω του git clone το επίσημο αποθετήριο της liblsl για Matlab<sup>6</sup>.
  - Σε περίπτωση που κατεβάσατε το αρχείο σε μορφή zip, αυτό πρέπει να αποσυμπιεστεί.
2. Προσθέστε την τελευταία έκδοση των δυαδικών αρχείων της liblsl στο μονοπάτι liblsl-Matlab/bin/
  - Επιλογή 1: Δεν απαιτείται κάποια ενέργεια. Τα αρχεία που χρειάζονται θα κατέβουν μόνα τους. Σημειώνεται πως υπάρχει περίπτωση αυτό το βήμα να αποτύχει. Ακόμα και αν κατέβουν τα αρχεία υπάρχει περίπτωση να παρουσιαστεί κάποιο σφάλμα.
  - Επιλογή 2: Κατεβάστε την pre-compiled έκδοση της liblsl<sup>7</sup>
3. Δημιουργήστε τα .mex αρχεία.
  - Στο Matlab, αλλάξτε το τρέχον μονοπάτι εργασίας στο root του αποθετηρίου (e.g. liblsl-Matlab/) και εκτελέστε το build\_mex
  - Από το command line εκτελέστε την εντολή `matlab -nodesktop -nosplash -r 'build_mex'`
  - Σε περίπτωση που δεν υπάρχει εγκατεστημένος κάποιος compiler πρέπει να εγκαταστήσετε έναν. Στα Windows, χρειάζεται απλά να εγκαταστήσετε το Visual Studio.

<sup>6</sup> <https://github.com/labstreaminglayer/liblsl-Matlab>

<sup>7</sup> <https://github.com/sccn/liblsl/releases>

## 4 ΕΓΚΑΤΑΣΤΑΣΗ

Στην ενότητα αυτή παρουσιάζονται οι προδιαγραφές του συστήματος και τα πακέτα τα οποία είναι απαραίτητα να εγκατασταθούν για την καταγραφή συγχρονισμένων ροών δεδομένων από τους διαθέσιμους αισθητήρες αλλά και των γεγονότων (events) που λαμβάνουν χώρα κατά τη διάρκεια της πειραματικής διαδικασίας. Τέλος, παρέχονται αναλυτικές οδηγίες για την εγκατάσταση και την ορθή χρήση των συγκεκριμένων πακέτων.

### 4.1 ΠΡΟΔΙΑΓΡΑΦΕΣ

Ένα από τα κυριότερα στοιχεία του υπό-σχεδίαση συστήματος αποτελεί η επεξεργασία των καταγεγραμμένων σημάτων τόσο του ET όσο και του EEG, η οποία σε αρκετές περιπτώσεις μπορεί να είναι ιδιαίτερα κοστοβόρα. Πράγματι, όλα τα σύγχρονα συστήματα παρακολούθησης ματιών βασίζονται σε μία ή περισσότερες κάμερες υπέρυθρων που καταγράφουν ροές βίντεο σε πραγματικό χρόνο από τα μάτια του χρήστη-συμμετέχοντα. Αυτές οι ροές βίντεο συνήθως μεταφέρονται σε κεντρικό υπολογιστή μέσω σύνδεσης υψηλής ταχύτητας (USB 2.0 ή 3.0, Firewire, GigE ή CamLink) στην οποία εκτελείται το λογισμικό ανίχνευσης και παρακολούθησης ματιών. Αυτό το λογισμικό αποτελείται από ένα σύνολο πολύπλοκων και συνήθως ιδιόκτητων (δηλαδή ανήκουν στην εταιρία που παράγει τον ET) αλγορίθμων μηχανικής όρασης που εκτελούνται, στην μεγάλη τους πλειοψηφία, στον κεντρικό υπολογιστή - παρόλο που υπάρχουν μερικές συσκευές ET οι οποίες προσφέρουν ενσωματωμένες δυνατότητες επεξεργασίας στην ίδια τη συσκευή.

Ομοίως, τα συστήματα εγγραφής EEG καταγράφουν την ηλεκτρική δραστηριότητα του εγκεφάλου μέσω ενός συνόλου ηλεκτροδίων τοποθετημένων σε διαφορετικά σημεία στο τριχωτό της κεφαλής. Αυτή η δραστηριότητα εν συνεχεία ενισχύεται μέσω ενός ενισχυτή που δημιουργεί, αυτό που συνήθως αναφέρεται ως ακατέργαστα δεδομένα, δηλαδή, σήματα δειγμάτων πλήρους εύρους ζώνης. Ωστόσο, προκειμένου αυτά τα σήματα να καταστούν χρησιμοποιήσιμα σε ένα σύστημα BCI, χρησιμοποιείται μια συγκεκριμένη ακολουθία βημάτων προεπεξεργασίας όπως αποθορυβοποίηση, αφαίρεση σημάτων μη-εγκεφαλικής προέλευσης, φιλτράρισμα σε συγκεκριμένο εύρος ζώνης και φασματική ανάλυση. Εάν θεωρήσουμε ότι σε ένα σύστημα BCI οι περισσότερες από αυτές τις λειτουργίες θα πρέπει να εκτελούνται σε πραγματικό χρόνο στα πολυδιάστατα εγκεφαλικά σήματα (συνήθως περιλαμβάνουν από 8 έως 256 ηλεκτρόδια), γίνεται εμφανές ότι οι αλγόριθμοι που είναι απαραίτητοι για την εξαγωγή υψηλού επιπέδου διανοητικής λειτουργίας μπορεί να απαιτήσουν σημαντική ποσότητα υπολογιστικών πόρων.

Από την άλλη πλευρά, αισθητήρες που καταγράφουν ένα μόνο κανάλι (όπως το GSR και το HR που ενδεχομένως να συντελέσουν συμπληρωματικό ρόλο στα EEG και ET στο πλαίσιο του NeuroMkt) με αργά εξελισσόμενα σήματα, δεν απαιτούν επεξεργασία πολύπλοκων αλγορίθμων. Τέλος, η επεξεργασία των παραπάνω σημάτων θα γίνεται μέσω του Matlab και ως εκ τούτου ο υπολογιστής που θα επιλεγεί θα πρέπει να μπορεί να «υποστηρίξει» το συγκεκριμένο λογισμικό, κυρίως σε θέματα μνήμης και επεξεργαστικής ισχύος.

Η συστοιχία αισθητήρων που θα χρησιμοποιηθεί στο πλαίσιο του NeuroMkt (μία συσκευή EEG, μία ET και ενδεχομένως ακόμα ένας μονοκάναλος αισθητήρας) θα συνδέεται με έναν υπολογιστή, ενώ σε περίπτωση που απαιτείται η μετάδοση και οπτικοποίηση των διανοητικών λειτουργιών σε πραγματικό χρόνο στους τελικούς χρήστες (διαφημιστές), κρίνεται αναγκαία η αξιοποίηση ενός ακόμα υπολογιστικού συστήματος που θα φιλοξενεί το dashboard.

Με βάση τα παραπάνω, οι αρχικές προδιαγραφές ενός τέτοιου υπολογιστικού συστήματος, όπως παρουσιάζονται στον Πίνακα 2, είναι συντηρητικά "high end" με σκοπό να διασφαλιστεί η ομαλή λειτουργία όλων των αισθητήρων χωρίς να υπάρχουν συστηματικές διακοπές. Σημειώνεται ότι στην παρούσα φάση είναι πρακτικώς αδύνατο να καθοριστούν με ακρίβεια οι ελάχιστες απαιτήσεις του συστήματος, ωστόσο κάτι τέτοιο θα διαμορφωθεί επαρκώς, σε μεταγενέστερο στάδιο του έργου.



Πίνακας 2. Οι προτεινόμενες αρχικές προδιαγραφές για το υπολογιστικό σύστημα που θα φιλοξενεί το λογισμικό του NeuroMkt.

Χαρακτηριστικό	Ελάχιστες Προδιαγραφές	Προτεινόμενες Προδιαγραφές
CPU	AMD Ryzen 5 1600 / Intel Core i5 (7 <sup>th</sup> generation)	AMD Ryzen 5 3600/ Intel Core i5 (10 <sup>th</sup> generation)
GPU	GTX 1060	RTX 2060
RAM	16GB	32GB
Συνδεσιμότητα	USB3.0 interface	USB3.0 interface, LPT interface

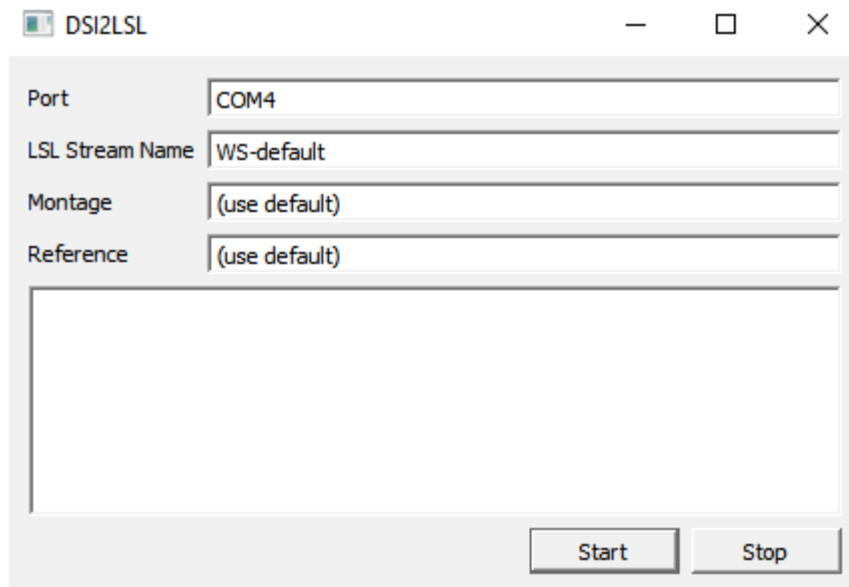
## 4.2 ΠΑΚΕΤΑ

Η λειτουργία του προτεινόμενου συστήματος απαιτεί τη χρήση συγκεκριμένων πακέτων για: (α) την καταγραφή των σημάτων από τους αισθητήρες, (β) την καταγραφή χρονικών σημάνσεων και γεγονότων κατά τη διάρκεια της διεξαγωγής της πειραματικής διαδικασίας, και (γ) το συγχρονισμό όλων των παραπάνω.

Ξεκινώντας από την καταγραφή των σημάτων και πιο συγκεκριμένα, αυτών του EEG, απαιτείται η εγκατάσταση δύο διαφορετικών πακέτων. Το πρώτο είναι το **DSI-Streamer**, το οποίο διασφαλίζει τη διασύνδεση του υπολογιστή με την EEG συσκευή και τη ροή δεδομένων από τη συσκευή προς τον υπολογιστή (βλέπε Εικόνα 3). Αντίστοιχα το δεύτερο είναι το **DSI2LSL** το οποίο αποστέλλει τη ροή δεδομένων της συσκευής στο Lab Streaming Layer (LSL) (βλέπε Εικόνα 4). Το LSL, το οποίο έχει παρουσιαστεί εκτενώς στο πλαίσιο του Παραδοτέου Π3.1 είναι ένα σύστημα το οποίο εξασφαλίζει τη συγχρονισμένη καταγραφή των διαφόρων σημάτων.

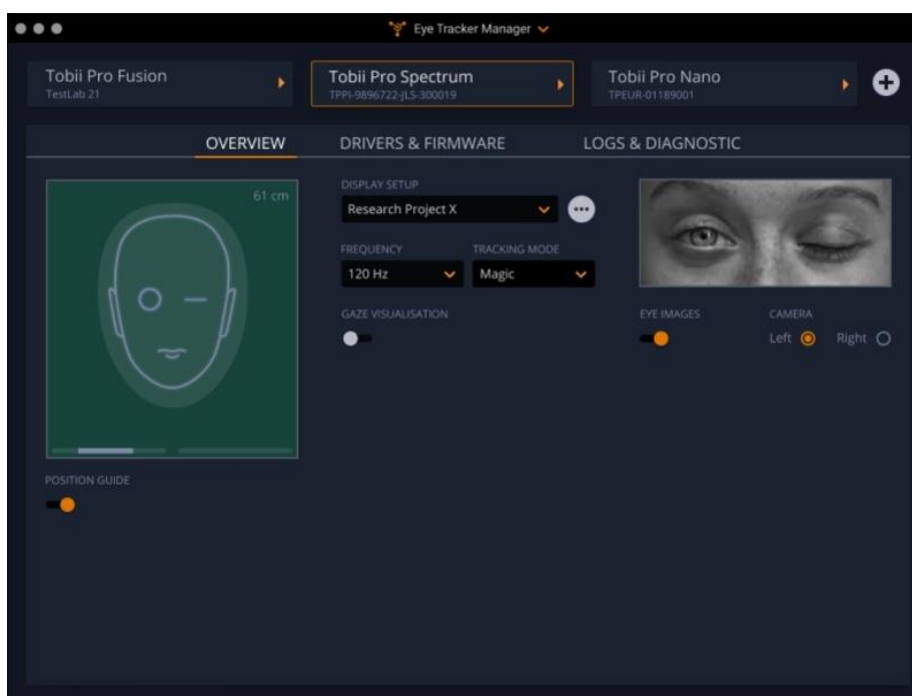


Εικόνα 3. Το user interface που προκύπτει από την εγκατάσταση του πακέτου DSI-Streamer.

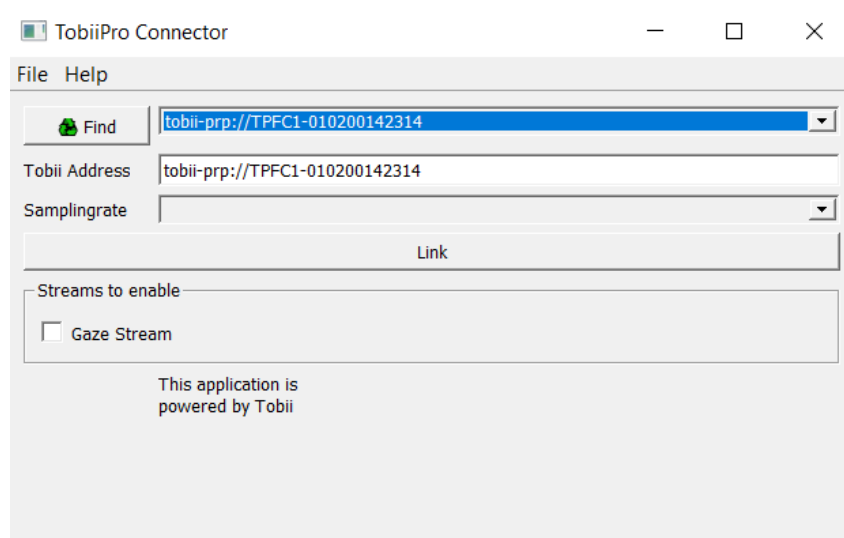


Εικόνα 4. Το user interface που προκύπτει από την εγκατάσταση του πακέτου DSI2LSL

Αντίστοιχα και για την καταγραφή των ET σημάτων χρειάζονται δύο πακέτα, με το πρώτο και πάλι να συμβάλει στη διασύνδεση μεταξύ του ET και του υπολογιστή και το δεύτερο να δημιουργεί δίαυλο επικοινωνίας με το LSL. Αναλυτικότερα, το πρώτο πακέτο είναι το **Eye Tracker Manager** (βλέπε Εικόνα 5), το οποίο πέρα από τη σύνδεση των δύο συσκευών χρησιμοποιείται για τη σωστή τοποθέτηση του υποκειμένου σε σχέση με τη συσκευή ET αλλά και τη διαδικασία του calibration, την προσαρμογή δηλαδή της ET συσκευής στο εκάστοτε άτομο το οποίο τη χρησιμοποιεί. Το δεύτερο πακέτο, το οποίο απεικονίζεται στην Εικόνα 6, ονομάζεται **TobiiProLSL**, και είναι αυτό που «μεταφέρει» τη ροή δεδομένων της συσκευής ET στο LSL.



Εικόνα 5. Το user interface που προκύπτει από την εγκατάσταση του πακέτου Eye Tracker Manager.



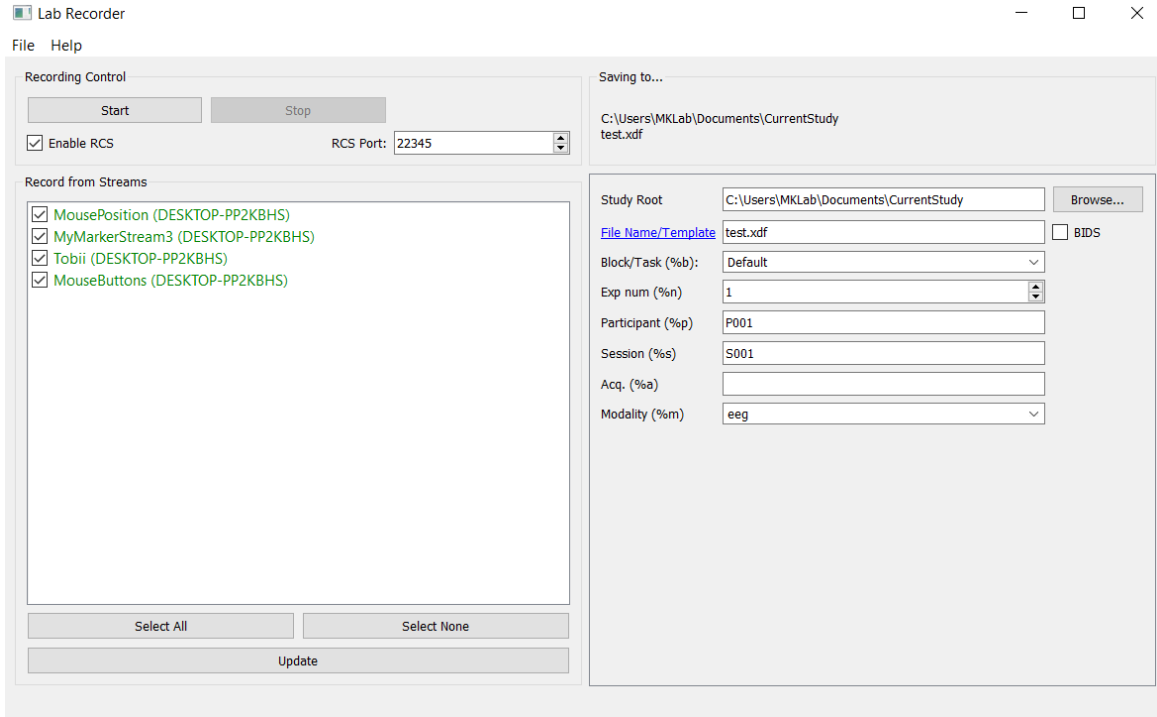
Εικόνα 6. Το user interface που προκύπτει από την εγκατάσταση του πακέτου TobiiProLSL.

Τα επόμενα δύο πακέτα αφορούν την καταγραφή χρονικών σημάνσεων για συγκεκριμένα γεγονότα που λαμβάνουν χώρα κατά τη διάρκεια εκτέλεσης του πειράματος και την αποστολή της σχετικής πληροφορίας στο LSL. Το πρώτο είναι το **Matlab LSL** και αφορά το πειραματικό πρωτόκολλο και την καταγραφή τόσο των χρονικών στιγμών εμφάνισης των οπτικών ερεθισμάτων, είτε πρόκειται για εικόνες είτε για βίντεο, όσο και του τύπου αυτών, με τα συγκεκριμένα στοιχεία να αποστέλλονται στο LSL. Το δεύτερο είναι το **Mouse LSL** (βλέπε Εικόνα 7) και καταγράφει τις κινήσεις του ποντικιού και τα κλικ που κάνει το άτομο που συμμετέχει στην πειραματική διαδικασία, στοιχεία τα οποία και πάλι αποστέλλονται στο LSL.



Εικόνα 7. Το παράθυρο που εμφανίζεται κατά τη χρήση του πακέτου Mouse LSL.

Το τελευταίο πακέτο είναι το Lab Recorder (βλέπε Εικόνα 8), το οποίο καταγράφει όλα τα προαναφερθέντα σήματα έχοντας κοινή χρονική αφετηρία. Η χρήση του συγκεκριμένου πακέτου είναι εξαιρετικής σημασίας καθώς με τον τρόπο αυτό μπορούμε να εξασφαλίσουμε ότι όλες οι μετέπειτα αναλύσεις θα μπορούν να γίνουν σε κοινό χρονικό άξονα.



Εικόνα 8. Το user interface του πακέτου Lab Recorder.

#### 4.3 ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ ΚΑΙ ΔΙΑΣΥΝΔΕΣΗΣ

Σε αυτή την υπο-ενότητα παρουσιάζονται αναλυτικά οι οδηγίες διασύνδεσης για τα πακέτα όπως αυτά παρουσιάστηκαν στην προηγούμενη ενότητα.

**DSI-Streamer:** Η Wearable Sensing, εταιρία κατασκευής της EEG συσκευής, είναι αυτή που παρέχει το συγκεκριμένο λογισμικό. Ουσιαστικά πρόκειται για ένα εκτελέσιμο αρχείο και ως εκ τούτου δεν απαιτείται η εγκατάσταση του στο συγκεκριμένο υπολογιστικό σύστημα. Το μόνο που απαιτείται από το χρήστη είναι το «άνοιγμά» του εκτελέσιμου αρχείου. Στη συνέχεια θα πρέπει να επιλεγεί η αντίστοιχη θήρα σύζευξης με τον τρόπο που περιγράφεται παρακάτω και να πατηθεί το κουμπί «connect» για την έναρξη της ροής δεδομένων από τη συσκευή EEG στον υπολογιστή. Πριν από το άνοιγμα του συγκεκριμένου αρχείου, απαιτούνται κάποιες επιπλέον, προπαρασκευαστικές, ενέργειες από το χρήστη. Αρχικά, πρέπει να εγκατασταθούν οι οδηγοί (drivers) της EEG συσκευής στον υπολογιστή με τον οποίο θέλουμε να συνδέσουμε τη συσκευή. Στη συνέχεια και αφού έχει ενεργοποιηθεί η συσκευή EEG θα πρέπει να γίνει η σύζευξη μεταξύ αυτής και του υπολογιστή μέσω Bluetooth. Η συγκεκριμένη διαδικασία είναι ιδιαίτερα απλή καθώς τα απαιτούμενα βήματα είναι ίδια με αυτά της σύζευξης μιας οποιασδήποτε συσκευής. Τέλος, ο χρήστης καλείται να σημειώσει τη θήρα σύζευξης (COM port), στοιχείο το οποίο απαιτείται κατά την ενεργοποίηση του DSI-Streamer. Τέλος, είναι σημαντικό να αναφερθεί ότι στην περίπτωση που δεν υπάρχει Bluetooth στον υπολογιστή όπου γίνεται η εγκατάσταση, θα πρέπει να τοποθετηθεί σε αυτόν κάποιος προσαρμογέας για να επιτευχθεί η διασύνδεση υπολογιστή-συσκευής.

**DSI2LSL:** Αντίστοιχα με το DSI-Streamer και το DSI2LSL είναι ένα εκτελέσιμο αρχείο, το οποίο κατά την εκτέλεσή του εκκινεί το user interface που φαίνεται στην Εικόνα 4. Στη συνέχεια, ο χρήστης πρέπει να επιλέξει εκ νέου τη σωστή θήρα σύζευξης, και να πατήσει το κουμπί «start» ώστε ξεκινήσει η ροή δεδομένων στο LSL. Τέλος, το LSL Stream Name, είναι το όνομα της ροής δεδομένων του EEG όπως αυτό θα εμφανίζεται στο LSL.

**Eye Tracker Manager:** Ο Eye Tracker Manager είναι ένα εκτελέσιμο πακέτο το οποίο παρέχεται από τον κατασκευαστή του ET. Η εκτέλεση του συγκεκριμένου πακέτου έχει ως αποτέλεσμα την εμφάνιση ενός user interface, μέσω του οποίου γίνονται όλα τα προπαρασκευαστικά βήματα πριν από την καταγραφή των δεδομένων. Κατά το πρώτο βήμα, ο Eye Tracker Manager εντοπίζει τη συσκευή ET (η οποία συνδέεται μέσω USB) και το μόνο που απαιτείται από το χρήστη είναι η σχετική επιβεβαίωση. Στη συνέχεια, εφόσον η θέση του ατόμου που θα χρησιμοποιήσει την ET συσκευή είναι η κατάλληλη, εκτελείται η διαδικασία του calibration.

**TobiiProLSL:** Το συγκεκριμένο πακέτο προκύπτει από τη διαδικασία μεταγλώττισης όπως αυτή παρουσιάζεται στην Ενότητα 3, η οποία έχει ως αποτέλεσμα το σχετικό εκτελέσιμο αρχείο. Είναι σημαντικό να σημειωθεί εδώ ότι, για τη λειτουργία του συγκεκριμένου πακέτου είναι απαραίτητη η ύπαρξη της Qt στο υπολογιστικό σύστημα. Η Qt είναι μία cross-platform εφαρμογή η οποία έχει δημιουργηθεί με τη χρήση της γλώσσας προγραμματισμού C++ , ενώ οι υποστηριζόμενες πλατφόρμες περιλαμβάνουν μεταξύ άλλων Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry και Sailfish OS. Η εκτέλεση του συγκεκριμένου πακέτου έχει ως αποτέλεσμα το user interface που παρουσιάζεται στην Εικόνα 5. Στη συνέχεια, ο χρήστης εφόσον έχει ήδη ανοιχτό το παράθυρο του Eye Tracker Manager, πατάει το κουμπί «Find» προκειμένου να γίνει η διασύνδεση της ET συσκευής με το LSL. Αν η διαδικασία έχει γίνει σωστά, εμφανίζεται το όνομα της συσκευής, ενώ στο τελευταίο βήμα ο χρήστης πρέπει να πατήσει το κουμπί «Link» για να ολοκληρωθεί η διασύνδεση.

**Matlab LSL:** Σε αυτή την περίπτωση, ουσιαστικά δημιουργείται μία ροή δεδομένων η οποία πηγάζει από το περιβάλλον του Matlab, το οποίο έχει χρησιμοποιηθεί για την προβολή των οπτικών ερεθισμάτων. Ως εκ τούτου, μία σειρά από εντολές όπως αυτές παρουσιάζονται στο παρακάτω κομμάτι κώδικα πρέπει να εκτελεστούν για την ενεργοποίηση της συγκεκριμένης ροής και την αποστολή της σχετικής πληροφορίας στο LSL. Όπως φαίνεται από το κομμάτι κώδικα, ο χρήστης θα πρέπει να προκαθορίσει το όνομα της ροής δεδομένων (π.χ. MyMarkerStream3) και στη συνέχεια να αρχικοποιήσει τη ροή δεδομένων με τη χρήση της συνάρτησης *lsl\_outlet*. Έπειτα, κάθε φορά που υπάρχει ανάγκη για καταχώρηση κάποιας σήμανσης καλείται η συνάρτηση *push\_sample* με βάση το επιλεγθέν outlet για την αποστολή της σχετικής πληροφορίας στο LSL (π.χ. *push\_sample('Left Arrow')*). Τέλος, για να λειτουργήσουν όλα τα παραπάνω θα πρέπει στο path του Matlab να υπάρχει η βιβλιοθήκη *liblsl-Matlab*.

```
%% Matlab Script for sending markers to LSL
lib = lsl_loadlib();
info =
lsl_streaminfo(lib, 'MyMarkerStream3', 'Markers', 1, 0, 'cf_float32', 'myuniqueid
23443');
outlet = lsl_outlet(info);
outlet.push_sample('Left Arrow');
```

**Mouse LSL:** Το Mouse LSL είναι ένα εκτελέσιμο αρχείο το οποίο κατά την εκτέλεση του εμφανίζει ένα παράθυρο όπως αυτό που φαίνεται στην Εικόνα 7, όπου σε περίπτωση επιτυχούς σύνδεσης του ποντικιού με το LSL εμφανίζονται τα μηνύματα «Mouse successfully hooked» και «Started streaming data».

**Lab Recorder:** Το Lab Recorder είναι ένα εκτελέσιμο αρχείο το οποίο κατά την εκκίνηση του εμφανίζει το user interface, το οποίο απεικονίζεται στην Εικόνα 8. Στο αριστερό τμήμα του interface εμφανίζονται όλες οι ενεργές ροές δεδομένων και ο χρήστης καλείται να επιλέξει ποιες από αυτές θέλει να καταγράψει κάνοντας κλικ στο αντίστοιχο checkbox. Στο συγκεκριμένο παράδειγμα (βλέπε Εικόνα 8), οι επιλεγμένες ροές δεδομένων είναι αυτές του ET (Tobii), του Matlab (MyMarkerStream3) και αυτές του ποντικιού (MousePosition και MouseButtons). Στη συνέχεια, ο χρήστης πρέπει να επιλέξει το φάκελο όπου θα αποθηκευτεί η συγκεκριμένη καταγραφή καθώς και το όνομα του αρχείου. Το τελευταίο βήμα αφορά το πάτημα του κουμπιού «Start» για να ξεκινήσει η καταγραφή των ροών δεδομένων. Τέλος, σημειώνεται ότι ο τύπος του αρχείου της καταγραφής είναι .xdf και ως εκ τούτου για τη

μεταφόρτωσή του σε περιβάλλον Matlab είναι απαραίτητη η βιβλιοθήκη *xdf-Matlab-master* η οποία θα πρέπει να τοποθετηθεί στο αντίστοιχο path του Matlab.

#### 4.4 ΟΡΓΑΝΩΣΗ ΚΑΙ ΧΡΗΣΗ ΚΩΔΙΚΑ ΣΤΟ GITHUB

Το σύνολο των πακέτων και των προγραμμάτων τα οποία παρουσιάστηκαν στις προηγούμενες ενότητες βρίσκονται στο αποθετήριο του NeuroMkt<sup>8</sup> που φιλοξενείται στο Github. Επιπρόσθετα, στο συγκεκριμένο αποθετήριο είναι διαθέσιμα τόσο ο Live Viewer όσο και το Dashboard τα οποία περιγράφονται στο Κεφάλαιο 5. Ο πιο εύκολος τρόπος για να χρησιμοποιήσει κανείς τον κώδικα του αποθετηρίου είναι να αντιγράψει το κυρίως project (cloning process) του ενδιαμέσου λογισμικού ακολουθώντας τα βήματα τα οποία περιγράφονται παρακάτω:

##### # Cloning GitHub Repository

```
[1] git clone --recursive --submodules https://github.com/NeuroMkt/Middleware.git  
[2] git clone https://github.com/NeuroMkt/Neuromarketing_Toolkit.git
```

Η εντολή [1] θα έχει ως αποτέλεσμα τη λήψη όλων των ενοτήτων (modules) του project, ενώ αντίστοιχα η εντολή [2] αφορά τη λήψη μόνο των Live Viewer και Dashboard. Η λήψη του project θα έχει ως αποτέλεσμα τη δημιουργία των παρακάτω φακέλων στο σύστημα αρχείων του υπολογιστή:

- **Φάκελος App-TobiiStreamEngine**, περιέχει τα απαραίτητα εργαλεία για τη δημιουργία εκτελέσιμου αρχείου με σκοπό τη διασύνδεση της συσκευής ET με το LSL.
- **Φάκελος dsi2lsl**, περιλαμβάνει τα εργαλεία για την ενσωμάτωση της συσκευής EEG, και πιο συγκεκριμένα της DSI-24, στο LSL.
- **Φάκελος App-Input**, περιέχει τα στοιχεία τα οποία είναι απαραίτητα για την καταγραφή των δεδομένων του ποντικιού από το LSL.
- **Φάκελος liblsl-Matlab**, περιέχει την διεπαφή της liblsl για το περιβάλλον του Matlab.
- **Φάκελος LabRecorder**, περιέχει την εφαρμογή του Lab Recorder, η οποία καταγράφει όλες τις διαθέσιμες ροές δεδομένων έχοντας κοινή δεικτοδότηση.
- **Φάκελος LiveViewer**, περιέχει τον MATLAB κώδικα για την online αξιολόγηση των σημάτων που καταγράφονται από τους αισθητήρες.
- **Φάκελος Dashboard**, περιέχει όλα εκείνα τα στοιχεία που είναι απαραίτητα για την ανάλυση σημάτων τα οποία συνδυάζονται προκειμένου να δημιουργηθεί ένα γραφικό περιβάλλον που απευθύνεται στους τελικούς χρήστες - διαφημιστές και έχει ως σκοπό την υποβοήθηση στην αξιολόγηση διαφημιστικών μηνυμάτων.

Οι τρεις πρώτοι φάκελοι είναι απαραίτητοι για την καταγραφή των διαφορετικών δεδομένων από το Lab Recorder. Παράλληλα, ο φάκελος LiveViewer, μπορεί να χρησιμοποιείται τόσο πριν από την έναρξη της καταγραφής αλλά και κατά τη διάρκεια αυτής, με στόχο την εξασφάλιση της υψηλότερης δυνατής ποιότητας των σημάτων (EEG και ET). Τέλος, εφόσον έχει γίνει η συγχρονισμένη καταγραφή των δεδομένων από τους αισθητήρες, μπορεί να χρησιμοποιηθεί ο φάκελος Dashboard, για την ανάλυση των καταγεγραμμένων σημάτων.

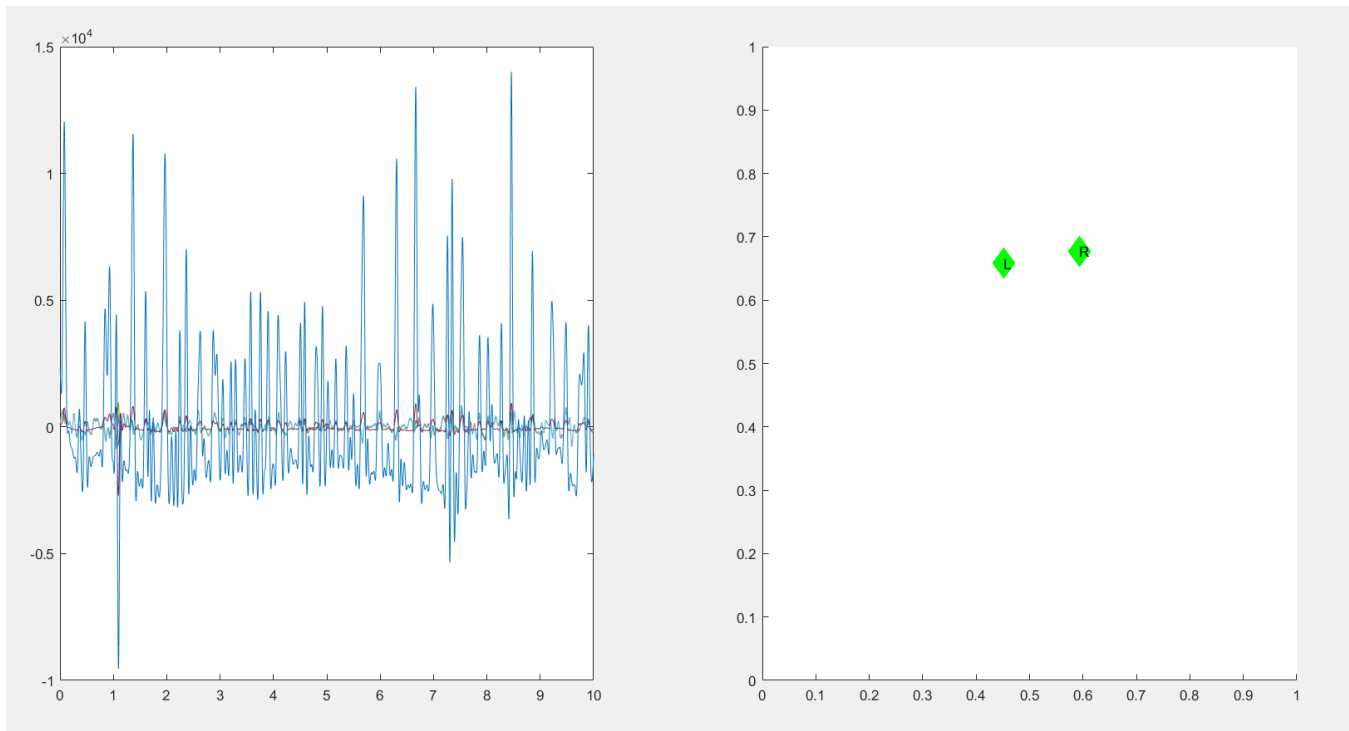
<sup>8</sup> <https://github.com/NeuroMkt/>

## 5 ΠΑΡΑΔΕΙΓΜΑΤΑ ΧΡΗΣΗΣ

Στην παρούσα ενότητα εργασίας παρουσιάζονται δύο χαρακτηριστικά παραδείγματα χρήσης του ενδιάμεσου λογισμικού του NeuroMkt. Το πρώτο παράδειγμα (Live Viewer) είναι ουσιαστικά ένα πρόγραμμα για οπτικοποίηση των σημάτων και αφορά την χρήση των σημάτων σε πραγματικό χρόνο. Το δεύτερο παράδειγμα αποτελεί μια πολύ πρώιμη έκδοση του Dashboard και λειτουργεί με σήματα που έχουν καταγραφεί σε προγενέστερο χρόνο με το Lab Recorder (Ενότητα 4.3).

### 5.1 LIVE VIEWER

Ο Live Viewer είναι ένα πρόγραμμα που έχει αναπτυχθεί στο πλαίσιο του έργου προκειμένου να δείχνει την ταυτόχρονη καταγραφή σημάτων EEG και ET σε πραγματικό χρόνο. Τα σήματα περνάνε πρώτα μέσα από την libls1, δεικτοδοτούνται χρονικά και στην συνέχεια δημιουργούνται τα κατάλληλα streams. Στη συνέχεια, ο Live Viewer κάνει resolve τα streams και οπτικοποιεί τα αντίστοιχα σήματα.



Εικόνα 9. Στιγμιότυπο από τον Live Viewer (πρόγραμμα που δείχνει σε πραγματικό τα συγχρονισμένα σήματα EEG και ET).

### 5.2 DASHBOARD

Μέσω ενός κατάλληλα διαμορφωμένου γραφικού περιβάλλοντος, ο χρήστης επιλέγει μια καταγραφή (δηλ. αρχείο .xdf όπως παράγεται από το Lab Recorder) για να την επεξεργαστεί. Καθώς τα σήματα προέρχονται από το Lab Recorder, ο συγχρονισμός τους είναι εξασφαλισμένος.

Το πρόγραμμα αυτό, στην παρούσα έκδοσή του, παρέχει συγχρονισμένες πληροφορίες σχετικά με τη φυσιολογία τόσο των ματιών όσο και του εγκεφάλου. Ακόμα και τα χαρακτηριστικά υψηλού επιπέδου (π.χ. approach-withdrawal index) παρέχονται σε κοινό χρονικό άξονα.



Εικόνα 10. Το γραφικό περιβάλλον μιας πολύ πρώιμης έκδοσης του Dashboard.



## 6 ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ ΚΑΙ ΣΥΧΝΕΣ ΕΡΩΤΗΣΕΙΣ

Παρακάτω περιγράφονται μερικά από τα πιο κοινότευα προβλήματα που απαντώνται κατά τη διάρκεια εκτέλεσης του λογισμικού που υποστηρίζει τη διασύνδεση των συσκευών:

**E1:** Το lsl stream που σχετίζεται με τον ET είναι κενό.

- Βεβαιωθείτε ότι η επιλογή “Stream Data” στην εφαρμογή TobiiPro-App είναι ενεργοποιημένη.

**E2:** Κάποιο (ή και όλα) από τα lsl streams δεν είναι διαθέσιμα.

- Βεβαιωθείτε ότι έχετε παράσχει την απαιτούμενη άδεια στο firewall ώστε να επιτρέψει τη μεταφορά των δεδομένων.

**E3:** Κάποια από τις εφαρμογές δεν ξεκινάει.

- Βεβαιωθείτε έχετε εκκινήσει την εφαρμογή με δικαιώματα διαχειριστή.

**E4:** Υπάρχει μεγάλη απόκλιση ανάμεσα στις πραγματικές συντεταγμένες του βλέμματος σας και σε αυτές που καταγράφονται.

- Βεβαιωθείτε ότι πρώτα κάνετε calibration και στην συνέχεια εκκινήσατε το TobiiPro-App.

- Σε περίπτωση που φοράτε γυαλιά πραγματοποιείστε calibration πολλαπλών σημείων.

**E5:** Μια εφαρμογή παρουσιάζει σφάλμα κατά την εκκίνησή της.

- Βεβαιωθείτε ότι έχετε ακολουθήσει ενδελεχώς τις παρεχόμενες οδηγίες.

- Αν η εφαρμογή υποστηρίζει γραφικό περιβάλλον, βεβαιωθείτε ότι η Qt βρίσκεται στο *system path*.

### 6.1 ΠΡΟΣΘΗΚΗ ΝΕΟΥ ΑΙΣΘΗΤΗΡΑ

Μια από τις πιο διαδομένες ερωτήσεις που συναντώνται σε αντίστοιχα συστήματα σχετίζεται με την επεκτασιμότητα του συστήματος προκειμένου να υποστηρίζει έναν νέο/διαφορετικό αισθητήρα. Παρακάτω παρέχονται οδηγίες προκειμένου να προγραμματιστεί ένα plugin που να διαμοιράζει τα δεδομένα μέσω της liblsl.

Οι οδηγίες δίνονται σε γλώσσα C/C++ καθώς πρόκειται για γλώσσα χαμηλού επιπέδου και συνεπώς υπάρχει μεγάλη πιθανότητα να μπορεί να επικοινωνεί κατευθείαν με τον εκάστοτε αισθητήρα. Εύκολα μπορεί κάποιος να συμπεράνει τις αντίστοιχες εντολές για μια άλλη γλώσσα, υψηλότερου επιπέδου, όπως το Matlab.

- Αρχικά προσθέστε τον C++ header της βιβλιοθήκης:

```
#include "lsl_cpp.h"
```

- Στη συνέχεια, δηλώστε ένα αντικείμενο stream info το οποίο ορίζει ότι ο αισθητήρας θα είναι διαθέσιμος με το όνομα “MyEEG” (το παράδειγμα αυτό αφορά έναν αισθητήρα EEG με 8 κανάλια και συχνότητα δειγματοληψίας 100 Hz, όπου κάθε κανάλι μεταφέρει δεδομένα τύπου float (32 bit)):

```
lsl::stream_info info("MyEEG", "EEG", 8, 100, lsl::cf_float32, "myuid");
```

- Κατόπιν, δημιουργείται ένα αντικείμενο stream outlet με ιδιότητες όπως ορίστηκαν στο αντικείμενο stream info:

```
lsl::stream_outlet outlet(info);
```

- Νέα δείγματα μπορούν να διαμοιράζονται μέσω του outlet όπως παρακάτω:

```
// Declare a buffer for our sample
```

```
float sample[8];

// Stream the samples into the outlet
while (true) {

// fictious function which returns a sample from my EEG device and
// copies it into our sample buffer
getSampleFromMySensor(sample);

// Push the sample into the LSL outlet
outlet.push_sample(sample);

}
```