Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο Περιφερειακής Ανάπτυξης
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΑΝΑΠΤΥΞΗΣ ΚΑΙ ΕΠΕΝΔΥΣΕΩΝ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΡΟΓΡΑΜΜΑΤΩΝ ΕΤΠΑ, ΤΣ & ΕΚΤ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Ανάπτυξη Μεθοδολογιών και Ενσωματωμένων Λύσεων Ασφάλειας για Τεχνολογίες Internet of Things σε ηλεκτρονικές Υπηρεσίες Υγείας – ΜΕΛΙΤΥ (Τ1ΕΔΚ-01958)

ΠΡΩΤΟΤΥΠΟ ΜΕΛΕΤΗΣ ΥΠΝΟΥ: ΕΝΣΩΜΑΤΩΜΕΝΟ ΣΥΣΤΗΜΑ ΚΑΙ ΛΟΓΙΣΜΙΚΟ

ΕΝΟΤΗΤΑ ΕΡΓΑΣΙΑΣ ΕΡΓΟΥ:	***ΕΕ2***
ΚΩΔΙΚΟΣ ΠΑΡΑΔΟΤΕΟΥ:	***Π2.2***
ΕΚΔΟΣΗ:	***Τελική***
ΚΑΤΑΣΤΑΣΗ:	***Υποβληθείσα***
ΗΜΕΡΟΜΗΝΙΑ ΟΛΟΚΛΗΡΩΣΗΣ:	**31/12/2019**
ΥΠΕΥΘΥΝΟΣ ΦΟΡΕΑΣ:	***Πανεπιστήμιο Ιωαννίνων***
ΣΥΜΜΕΤΕΧΟΝΤΕΣ ΦΟΡΕΙΣ	**ΟΛΥΜΠΙΟΝ**

Πίνακας Περιεχόμενων

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο Περιφερειακής Ανάπτυξης
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΑΝΑΠΤΥΞΗΣ ΚΑΙ ΕΠΕΝΔΥΣΕΩΝ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΡΟΓΡΑΜΜΑΤΩΝ ΕΤΠΑ, ΤΣ & ΕΚΤ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Πίνακας Πινάκων

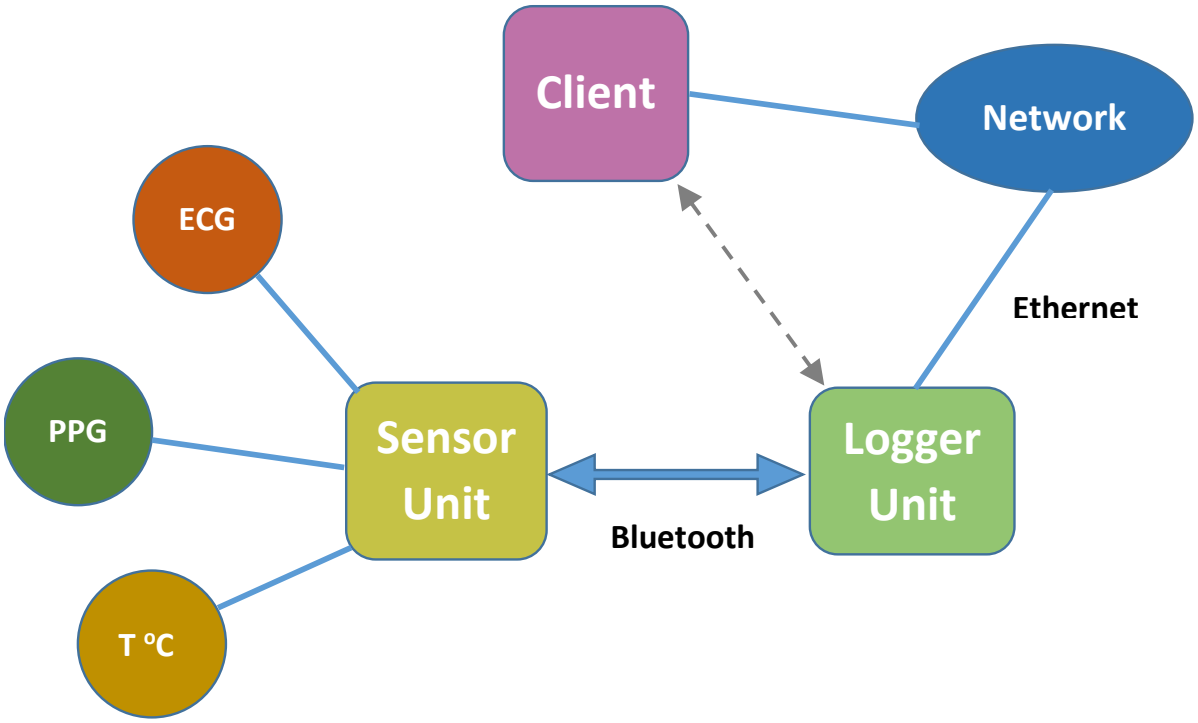
Πίνακας Εικόνων

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Ταμείο Περιφερειακής Ανάπτυξης
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΑΝΑΠΤΥΞΗΣ ΚΑΙ ΕΠΕΝΔΥΣΕΩΝ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΡΟΓΡΑΜΜΑΤΩΝ ΕΤΠΑ, ΤΣ & ΕΚΤ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Περιγραφή λειτουργία συστήματος

1. Περίληψη

Το σύστημα καταγραφής και παρακολούθησης βιοϊατρικών σημάτων αποτελείται από 2 μονάδες όπως διακρίνονται στο παρακάτω σχήμα, τη μονάδα αισθητήρων (sensor unit) η οποία χρησιμοποιείται για την λήψη των σημάτων με χρήση κατάλληλων αισθητήρων και τη μονάδα καταγραφής (logger unit). Οι δύο μονάδες επικοινωνούν με ασύρματη σύνδεση Bluetooth, ενώ ο χρήστης μπορεί να έχει πρόσβαση στα σήματα είτε σε πραγματικό χρόνο είτε offline (αρχείο καταγραφής) με κατάλληλο λογισμικό το οποίο επικοινωνεί με τη μονάδα logger μέσω δικτύου (TCP/IP Client).

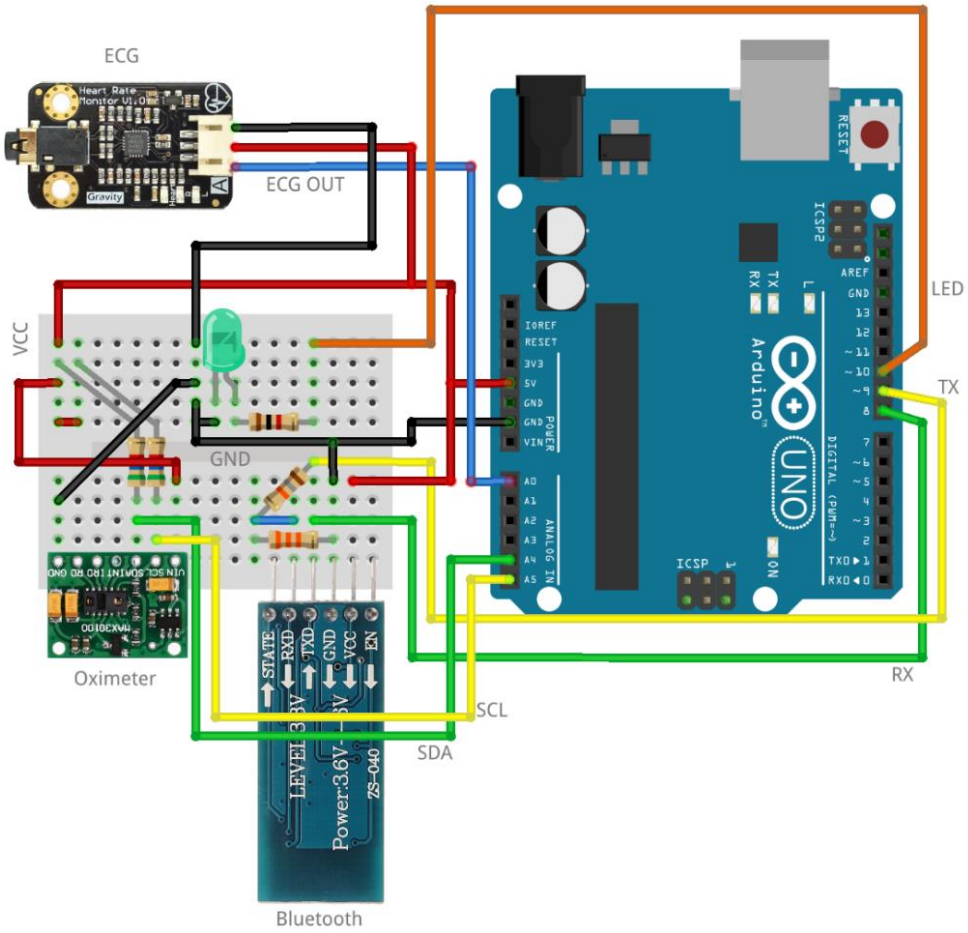


ΕΙΚΟΝΑ 1. ΣΧΗΜΑΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΛΕΙΤΟΥΡΓΙΑΣ ΣΥΣΤΗΜΑΤΟΣ

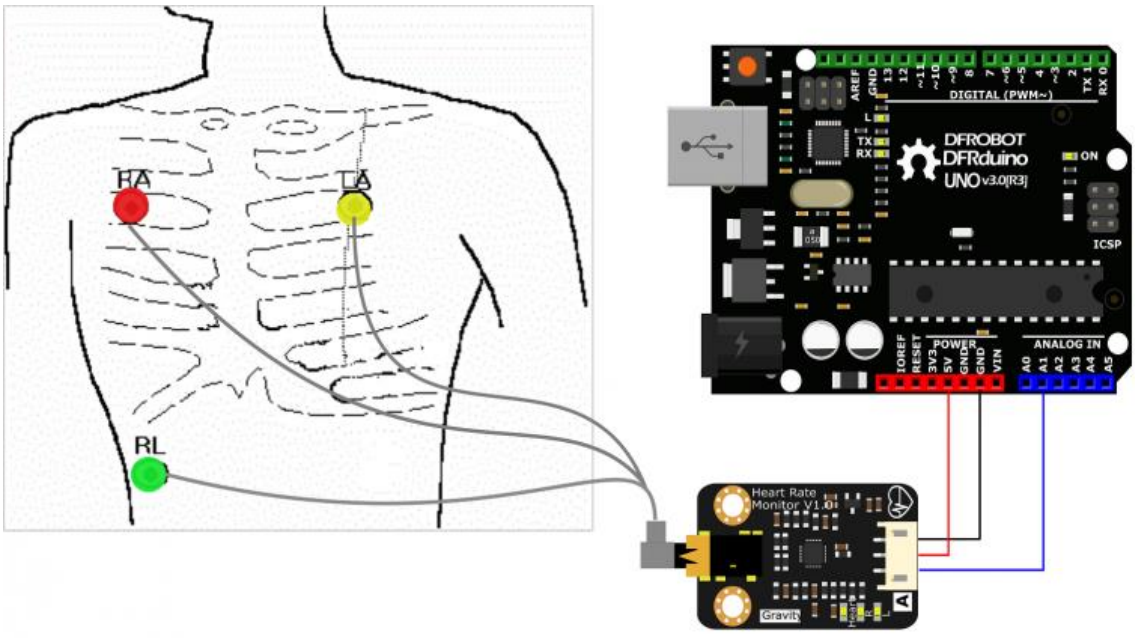
2. Μονάδα αισθητήρων

2.1. Γενικά

Το κύκλωμα αποτελείται από μία αναπτυξιακή πλακέτα Arduino UNO, μία συσκευή λήψης καρδιογραφήματος ενός καναλιού, ένα οξύμετρο και ένα πομποδέκτη Bluetooth για την επικοινωνία με το Logger (εικόνα 2). Η μέτρηση με το οξύμετρο πραγματοποιείται με τοποθέτηση του δαχτύλου απευθείας πάνω στον αισθητήρα (κόκκινο LED) του οξυμέτρου χωρίς πίεση. Για μέτρηση καρδιογραφήματος ακολουθείται η συνδεσμολογία της εικόνας 3 χρησιμοποιώντας τα ηλεκτρόδια επαφής με τον αντίστοιχο χρωματικό κώδικα.



ΕΙΚΟΝΑ 2. ΣΧΗΜΑΤΙΚΟ ΔΙΑΓΡΑΜΜΑ ΚΥΚΛΩΜΑΤΟΣ ΜΟΝΑΔΑΣ ΑΙΣΘΗΤΗΡΩΝ



ΕΙΚΟΝΑ 3. ΜΕΤΡΗΣΗ ΚΑΡΔΙΟΓΡΑΦΗΜΑΤΟΣ

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

2.2. Arduino UNO rev3

Το Arduino UNO (Arduino, 2019) είναι μια αναπτυξιακή πλακέτα ανοιχτού κώδικα με κεντρική μονάδα επεξεργασίας ένα 8-bit μικροελεγκτή αρχιτεκτονικής RISC, συγκεκριμένα το μοντέλο ATmega328p (Microchip Technology, 2019). Τα τεχνικά χαρακτηριστικά για το μικροελεγκτή και για την πλακέτα Arduino UNO φαίνονται στον πίνακα 1.

ΠΙΝΑΚΑΣ 1. ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ARDUINO UNO REV3

Μικροελεγκτής	ATmega328P
Τάση λειτουργίας	5V
Τάση ειδόδου	7-12V
Όρια τάσης ειδόδου	6-20V
Ακροδέκτες Εισόδου/Εξόδου (I/O)	14
Ακροδέκτες PWM	6
Αναλογικές είσοδοι	6 (10 bit)
Ένταση ρεύματος ακροδεκτών I/O	20 mA
Ένταση ρεύματος ακροδέκτη 5V	50 mA
Μνήμη Flash	32 KB (ATmega328P)
Μνήμη SRAM	2 KB (ATmega328P)
Μνήμη EEPROM	1 KB (ATmega328P)
Συχνότητα λειτουργίας	16 MHz
Περιφερειακά	8x 10-bit ADC 1x USART 1x SPI 1x I^2C

2.3. Συσκευή ασύρματης επικοινωνίας Bluetooth

Για την επικοινωνία μεταξύ της μονάδας καταγραφής και της μονάδας αισθητήρων γίνεται χρήση συσκευής Bluetooth και συγκεκριμένα το δημοφιλές μοντέλο HC-05 (Electrónica Estudio - Ingeniería Electrónica y Proyectos PICmicro®, 2010). Τα κυριότερα τεχνικά χαρακτηριστικά διακρίνονται στον πίνακα 2.

ΠΙΝΑΚΑΣ 2. ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΣΥΣΚΕΥΗΣ BLUETOOTH

Πρωτόκολλο	Bluetooth Specification v2.0+EDR
Συχνότητα	2.4GHz ISM band
Διαμόρφωση	GFSK(Gaussian Frequency Shift Keying)
Ισχύς εκπομπής	≤4dBm, Class 2
Εμβέλεια	10m
Ευαισθησία	≤-84dBm at 0.1% BER
Ασφάλεια	Authentication and encryption

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Προφίλ	Serial port (SPP)
Τροφοδοσία	+3.3VDC 50mA
Ρυθμοί μετάδοσης δεδομένων (baud rate)	9600, 19200, 38400, 57600, 115200, 230400, 460800

2.4. Συσκευή καρδιογραφήματος

Η μονάδα αισθητήρων διαθέτει μια συσκευή λήψης καρδιογραφήματος (ECG). Η πλακέτα που χρησιμοποιήθηκε είναι της εταιρίας DFRobot (DFRobot - Quality Arduino Robot IOT DIY Electronic Kit, 2019). Η πλακέτα βασίζεται στο ολοκληρωμένο κύκλωμα AD8232 από την Analog Devices (Analog Devices, 2018). Το ολοκληρωμένο είναι σε θέση va πάρει καρδιογράφημα ενός καναλιού (single-lead ECG) και χρησιμοποιεί τρία ηλεκτρόδια σύμφωνα με τη συνδεσμολογία της εικόνας 3. Ο πίνακας 3 δείχνει τα τεχνικά χαρακτηριστικά της συσκευής.

ΠΙΝΑΚΑΣ 3. ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΚΑΡΔΙΟΓΡΑΦΟΥ

Τάση εισόδου	3.3-6V (5V τυπική)
Τάση εξόδου	0-3.3V
Διασύνδεση	Αναλογική
Ένταση ρεύματος	<10mA
Κανάλια	1
Εύρος ζώνης	2kHz

2.5. Συσκευή οξυμέτρου

Η μονάδα αισθητήρων συμπληρώνεται με μία συσκευή με την οποία μπορεί κανείς να μετρήσει τον κορεσμό οξυγόνου στο αίμα (οξύμετρο). Η λειτουργία της βασίζεται στην τεχνική της φωτοπληθυσμογραφίας (Allen, 2007). Η πλακέτα που χρησιμοποιήθηκε είναι το μοντέλο RCWL-0530 (Peko's Library, 2018), έχει σαν κεντρική μονάδα το ολοκληρωμένο κύκλωμα MAX30100 της εταιρίας Maxim Integrated (Maxim Integrated - Analog, linear, & mixed-signal devices, 2014). Στον πίνακα 4 διακρίνονται κάποια από τα βασικά χαρακτηριστικά του οξυμέτρου.

ΠΙΝΑΚΑΣ 4. ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΟΞΥΜΕΤΡΟΥ

Τάση λειτουργίας	1.8V-5.5V
Διασύνδεση	I^2C
Ταχύτητα διαύλου	400 kHz
Ρυθμός δειγματοληψίας	50-1000 samples/sec

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Διακριτική ικανότητα A/D	14bit
Μέτρηση θερμοκρασία	Ναι

2.6. Λειτουργία

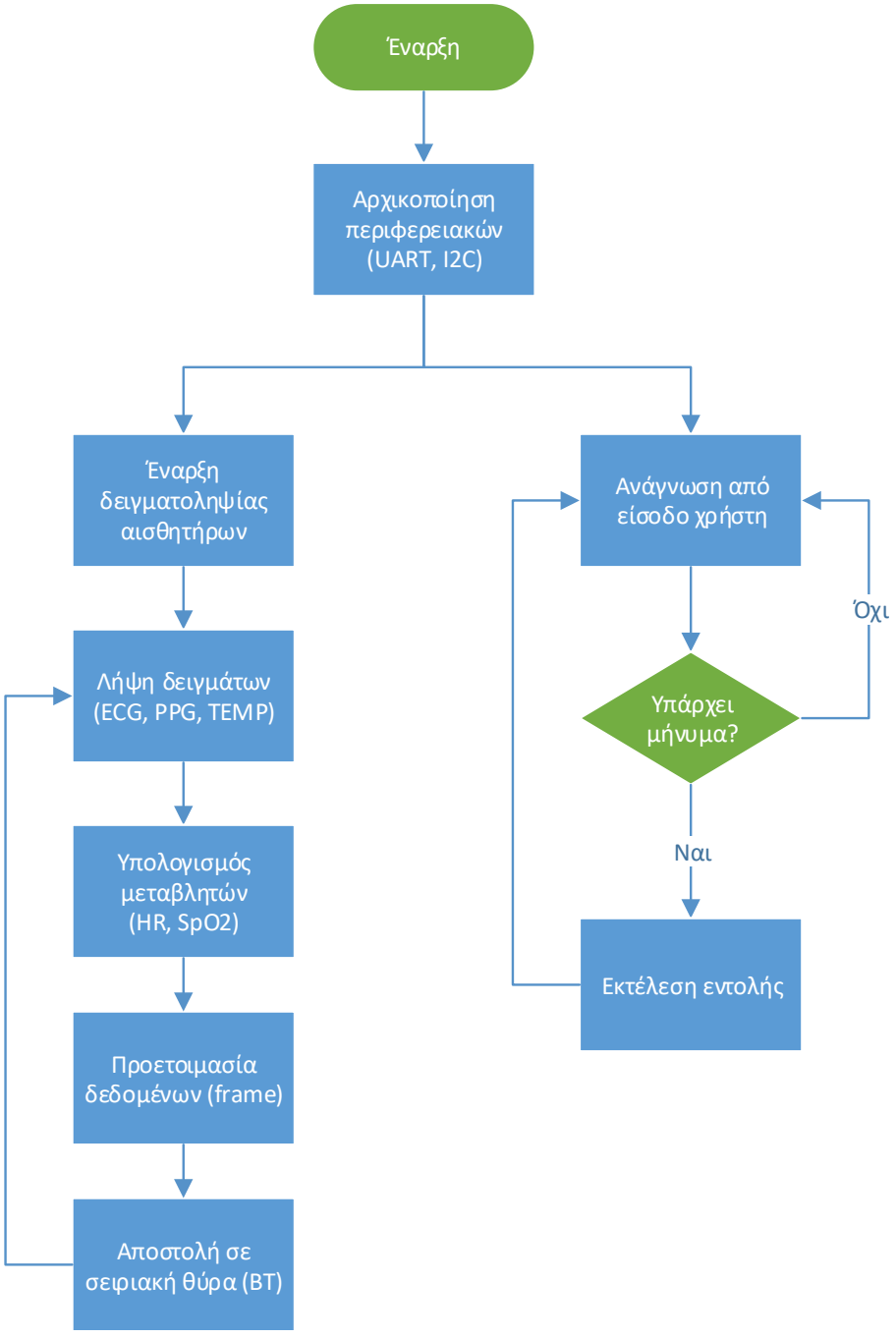
Οι περιφερειακές διασυνδέσεις για το Arduino είναι οι εξής:

- Για την επικοινωνία με το Bluetooth χρησιμοποιήθηκε σειριακή θύρα (UART) υλοποιημένη με λογισμικό (software serial port), στους ακροδέκτες 8 (RX) και 9 (TX) μέσω κατάλληλης βιβλιοθήκης. Ο ρυθμός μετάδοσης (baud) έχει τεθεί στα 57600 bps.
- Για τη λήψη καρδιογραφήματος χρησιμοποιείται αναλογική είσοδος (ADC) στο κανάλι εισόδου 0 του Arduino.
- Για την λήψη δειγμάτων οξυμέτρου (PPG) χρησιμοποιείται η θύρα I^2C. Από τον ίδιο αισθητήρα γίνεται λήψη και της θερμοκρασίας.
- Η ενσωματωμένη θύρα σειριακής επικοινωνίας (Hardware USART) του μικροελεγκτή αφιερώνεται στην αποσφαλμάτωση (debugging) και συνδέεται μέσω USB θύρας σε Η/Υ, ώστε να μπορεί ο χρήστης να λαμβάνει μηνύματα σχετικά με τη λειτουργία της μονάδας σε πραγματικό χρόνο.

Σκοπός αυτής της μονάδας είναι η περιοδική συλλογή δειγμάτων από τους αισθητήρες και η αποστολή σε πραγματικό χρόνο στη μονάδα καταγραφής. Για το σκοπό αυτό αναπτύχθηκε κώδικας σε γλώσσα C/C++ για Arduino και το αντίστοιχο διάγραμμα ροής των επιμέρους διεργασιών διακρίνεται στην εικόνα 4. Ο μικροελεγκτής προγραμματίζει το οξύμετρο να παίρνει δείγματα από τα δύο ενσωματωμένα LED (κόκκινο και υπέρυθρο, RED - IR) συνεχώς με ρυθμό 100 samples/sec. Συγχρονισμένα με αυτό το ρυθμό λαμβάνονται από το μικροελεγκτή δείγματα από τον καρδιογράφο (αναλογική είσοδος στο μικροελεγκτή) καθώς και από εσωτερικό θερμόμετρο του οξυμέτρου. Συνολικά τα σήματα που λαμβάνονται είναι 4 (ECG, PPG-RED, PPG-IR, Temp). Παράλληλα με τη λήψη των δειγμάτων το λογισμικό διαχείρισης του οξυμέτρου υπολογίζει τον κορεσμό οξυγόνου (SpO2) σε ποσοστό %, καθώς και τον καρδιακό ρυθμό (Heart-Rate, HR) σε παλμούς ανά λεπτό (bpm). Τα 6 συνολικά δείγματα συγκεντρώνονται και αποθηκεύονται προσωρινά. Ο πίνακας 5 δείχνει περιληπτικά τα δείγματα και το χώρο που καταλαμβάνουν στη μνήμη. Στη συνέχεια προετοιμάζονται και τακτοποιούνται σε συγκεκριμένη μορφή πακέτου ώστε να αποσταλούν σειριακά μέσω της σειριακής θύρας στην οποία είναι συνδεδεμένη η συσκευή Bluetooth. Η συσκευή Bluetooth αξιοποιεί την υπηρεσία σειριακής σύνδεσης (Serial Port Protocol, SPP) και εφόσον έχει συζευχθεί προηγουμένως αυτόματα με τη συσκευή καταγραφής (logger) στέλνει αυτόματα τα δεδομένα του πακέτου. Το πακέτο δεδομένων περιγράφεται στον πίνακα 6 και είναι συμβατό με την αναπτυξιακή πλακέτα βιο-ιατρικών σημάτων HealthyPi (ProtoCentral HealthyPi v3 3.1.0, 2019). Αποτελείται συνολικά από 27 bytes. Ο συνολικός ρυθμός δυαδικών δεδομένων είναι επομένως (27bytes/frame)*(100frames/sec)=2700bytes/sec το οποίο αντιστοιχεί σε ρυθμό baud rate σειριακής

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

θύρας (2700bytes/sec)*(10bits/byte)=27000bps, υπολογίζοντας τα 2 επιπλέον bits (start-stop) της UART.



ΕΙΚΟΝΑ 4. ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ ΤΗΣ ΜΟΝΑΔΑΣ ΑΙΣΘΗΤΗΡΩΝ

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

ΠΙΝΑΚΑΣ 5. ΤΑ ΔΕΙΓΜΑΤΑ ΑΠΟ ΤΟΥΣ ΑΙΣΘΗΤΗΡΕΣ ΑΝΑ ΠΑΚΕΤΟ

Όνομα	**Περιγραφή**	**Μήκος τιμής (bytes)**
ECG	Ψηφιακή αναπαράσταση της αναλογικής τιμής εξόδου του καρδιογράφου	2
PPG-IR	Τιμή που αντιστοιχεί στο υπέρυθρο LED του οξυμέτρου	4
PPG-RED	Τιμή που αντιστοιχεί στο κόκκινο LED του οξυμέτρου	4
Temp	Θερμοκρασίας σε ^{0}C	2
SpO2	Κορεσμός οξυγόνου (%)	1
HR	Καρδιακός ρυθμός (bpm)	1

ΠΙΝΑΚΑΣ 6. ΔΟΜΗ ΠΑΚΕΤΟΥ ΑΠΟΣΤΟΛΗΣ ΔΕΙΓΜΑΤΩΝ ΑΠΟ ΑΙΣΘΗΤΗΡΕΣ

Θέση Byte	**Τιμή**	**Περιγραφή**
00	0x0A	Αρχή πακέτου
01	0xFA	Αρχή πακέτου
02	-	-
03	-	-
04	Protocol version	(currently 0x02)
05-06	ECG Value	Signed int 16,LSB first
07-08	-	-
09-12	PPG IR Value	Signed int 32, LSB first
13-16	PPG Red Value	Signed int 32, LSB first
17-18	Temperature	Signed int 16, LSB first
19	-	-
20	SpO2	Unsigned int 8
21	Heart Rate	Unsigned int 8
22-24	Frame index	3 bytes, LSB first
25	0x00	Τέλος πακέτου
26	0x0B	Τέλος πακέτου

3. Μονάδα παρακολούθησης/καταγραφής

Η μονάδα παρακολούθησης/καταγραφής (logger) βασίζεται σε Raspberry Pi, και συγκεκριμένα χρησιμοποιήθηκε το μοντέλο Raspberry Pi 3 Model B+ (Rpi3B+) (Raspberry Pi, 2018). Τα σημαντικότερα τεχνικά χαρακτηριστικά δίνονται στον πίνακα 7. Με αυτό τον τρόπο η μονάδα αυτή είναι φορητή και αυτόνομη και δε χρειάζεται κάποια σύνδεση τερματικού ώστε να λειτουργήσει. Το λογισμικό υλοποιήθηκε σε γλώσσα προγραμματισμού C/C++ και είναι συμβατό με λειτουργικό σύστημα Linux όπως είναι και το λειτουργικό που χρησιμοποιήθηκε στην εν λόγω συσκευή (Raspbian Buster Lite)

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

(Raspberry Pi, 2019). Είναι επίσης εφικτό το ίδιο πρόγραμμα να λειτουργήσει σε οποιοδήποτε Η/Υ με λειτουργικό τύπου Linux. Έχει δοκιμαστεί σε υπολογιστεί με Ubuntu 18.04 32bit. Οι δύο απαιτήσεις συστήματος είναι να διαθέτει σύνδεση δικτύου (π.χ. Ethernet, WiFi) και σύνδεση Bluetooth. Η μονάδα καταγραφής αναλαμβάνει τα ακόλουθα:

- Συνεχή λήψη σημάτων από τη μονάδα αισθητήρων
- Αποθήκευση σημάτων τοπικά σε αρχείο
- Αποστολή των σημάτων στο χρήστη σε πραγματικό χρόνο

Η εικόνα 5 δείχνει το απλοποιημένο διάγραμμα ροής του προγράμματος που τρέχει στη μονάδα αυτή.

3.1. Λειτουργία

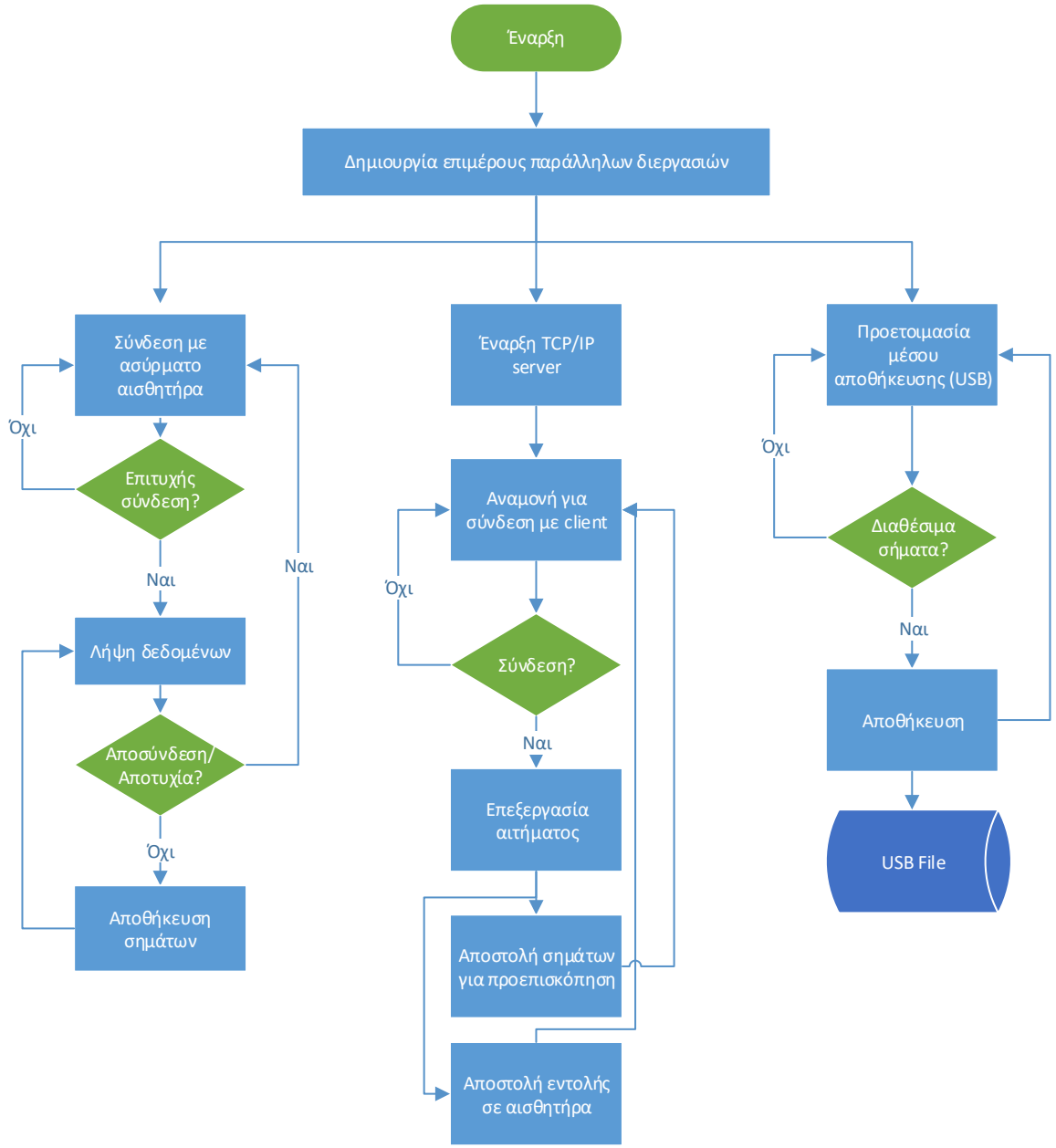
Συνδέουμε τη συσκευή (Rpi3B+) με το τροφοδοτικό και μετά από 1 λεπτό περίπου ξεκινά αυτόματα το πρόγραμμα. Εφόσον έχουμε δώσει τροφοδοσία και στη συσκευή αισθητήρων (sensor) οι 2 συσκευές συνδέονται ασύρματα αυτόματα, σε περίπτωση αποτυχίας (σβήσιμο, απομάκρυνση κ.λπ.) κάθε 5 δευτερόλεπτα ο logger επιχειρεί επανασύνδεση. Ο logger λειτουργεί σε debugging mode και τα βασικότερα μηνύματα λειτουργίας και σφαλμάτων τα τυπώνει σε κονσόλα σε οθόνη η οποία μπορεί να συνδεθεί προαιρετικά (η φορητή οθόνη 7" LCD συνδέεται στο raspberry με καλώδιο HDMI και USB τα οποία παρέχονται). Κάθε 1 λεπτό τα σήματα που έχουν υποθηκευθεί προσωρινά στη μνήμη, αποθηκεύονται σε αρχείο τοπικά (usb flash memory). Τα δείγματα που έχουν ληφθεί δομημένα σε πακέτα αποθηκεύονται με παρόμοιο τρόπο. Για κάθε πακέτο (frame) αφιερώνονται 26 bytes στα οποία περιλαμβάνονται τα σήματα καθώς και η χρονική στιγμή λήψης. Η μορφή όπως αποθηκεύονται στο αρχείο δίνεται στον πίνακα 8.

ΠΙΝΑΚΑΣ 7. ΤΕΧΝΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ RASPBERRY PI 3 MODEL B+

Επεξεργαστής	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Μνήμη	1GB LPDDR2 SDRAM
Λειτουργικό σύστημα	Raspbian
Συνδεσιμότητα	2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN
	Bluetooth 4.2, BLE
	Gigabit Ethernet over USB 2.0 (maximum throughput 300Mbps)
	4 × USB 2.0 ports
Είσοδοι / Έξοδοι γενικής χρήσης	40-pin GPIO header
Έξοδος βίντεο	HDMI

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Χώρος αποθήκευσης	Micro SD
Τροφοδοσία	5V/2.5A DC μέσω σύνδεσης micro USB



ΕΙΚΟΝΑ 5. ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ ΚΑΤΑΓΡΑΦΗΣ ΣΗΜΑΤΩΝ

ΠΙΝΑΚΑΣ 8. ΜΟΡΦΗ ΑΡΧΕΙΟΥ ΑΠΟΘΗΚΕΥΣΗΣ ΣΗΜΑΤΩΝ (ΑΝΑ ΠΑΚΕΤΟ ΔΕΔΟΜΕΝΩΝ)

Θέση byte	**Περιγραφή**	**Μήκος/Τύπος (Little Endian)**
0	Καρδιογράφημα (ECG)	4 (float)
4	Οξύμετρο (PPG IR LED)	4 (float)

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

8	Οξύμετρο (PPG RED LED)	4 (float)
12	Θερμοκρασία (Temp)	4 (float)
16	Κορεσμός οξυγόνου (SpO2)	1 (uint8_t)
17	Καρδιακός ρυθμός (HR)	1 (uint8_t)
18	Χρονοσήμανση (Timestamp)	8 (double)

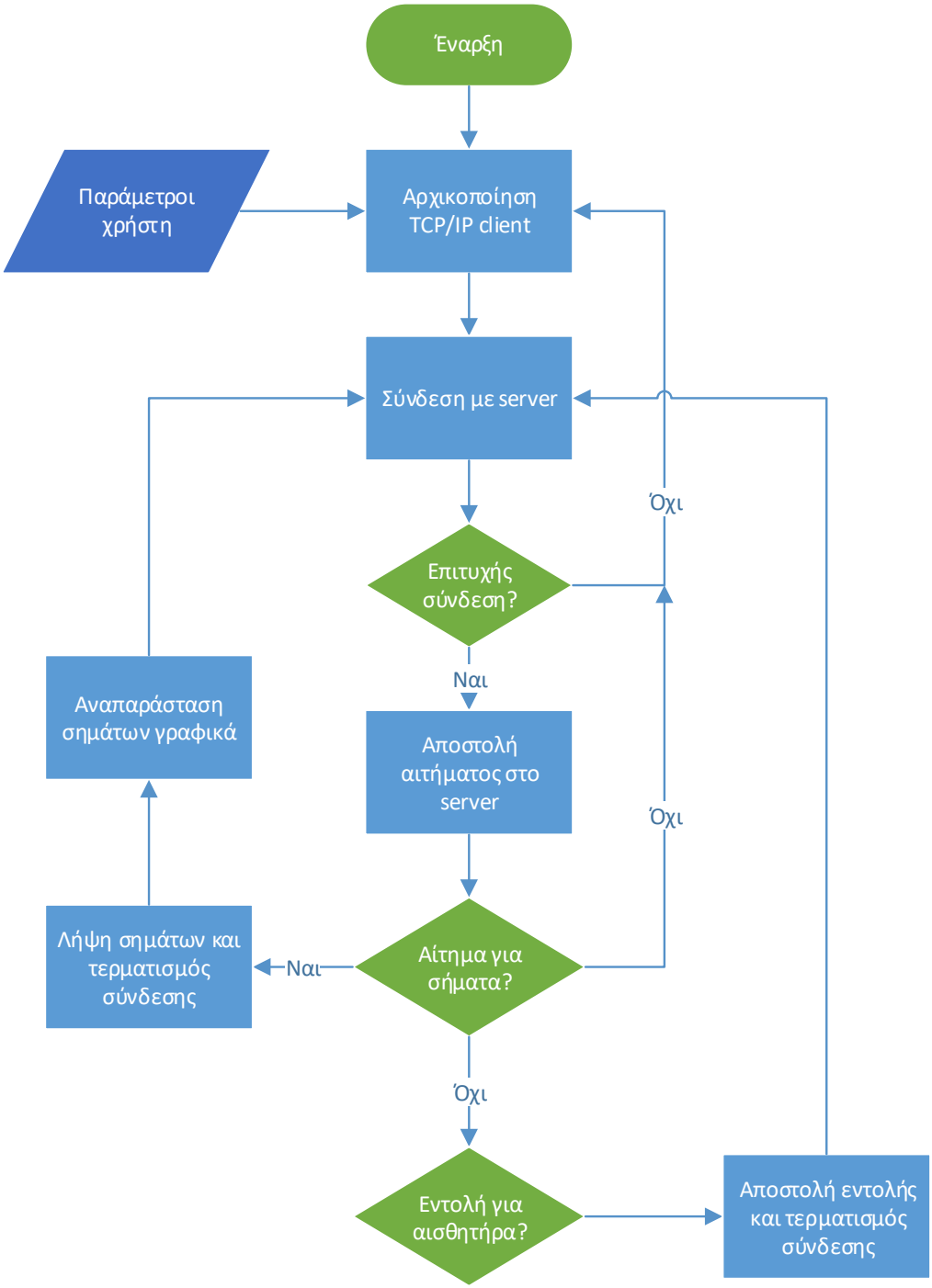
4. Λογισμικό παρακολούθησης

Το λογισμικό παρακολούθησης έχει υλοποιηθεί σε γλώσσα προγραμματισμού Matlab (MATLAB - MathWorks, 2019) και δίνει τη δυνατότητα στο χρήστη να παρακολουθήσει σε πραγματικό χρόνο τα σήματα που καταγράφονται, συγκεκριμένα τα στιγμιότυπα των τελευταίων 10 δευτερολέπτων της καταγραφής. Το πρόγραμμα αυτό αποτελεί τον client ο οποίος συνεργάζεται με τον server του logger και ανά τακτά χρονικά διαστήματα ζητά τα πιο πρόσφατα δεδομένα ώστε να τα αναπαραστήσει γραφικά. Ο client στέλνει εντολές στο logger που αποκωδικοποιούνται αναλόγως. Συγκεκριμένα υπάρχουν δύο εντολές. Με την πρώτη ο client ζητά τα δείγματα των σημάτων και με τη δεύτερη στέλνει μήνυμα προς την πλακέτα αισθητήρων ώστε να λειτουργήσει σαν κωδικοποιημένη εντολή προς κάποιον αισθητήρα. Οι δύο εντολές συγκεκριμένα είναι οι συμβολοσειρές "data" και "cmd1" αντίστοιχα και στέλνονται μέσω TCP/IP sockets. Με αυτό τον τρόπο ο χρήστης εκτός από δέκτης μπορεί να γίνει και πομπός μηνυμάτων. Στην περίπτωση αιτήματος από το server (logger) των πρόσφατων σημάτων ("data") ο logger απαντά με ακολουθία δυαδικών δεδομένων η οποία περιέχει την απαραίτητη πληροφορία. Η ακολουθία έχει μήκος 56 bytes και συγκεκριμένα η μορφή δίνεται στον παρακάτω πίνακα. Όλες οι τιμές μετατρέπονται σε μεταβλητές τύπου double που είναι η τυπικές μεταβλητές του Matlab. Στην εικόνα 6 δίνεται το διάγραμμα ροής του προγράμματος για τον client.

ΠΙΝΑΚΑΣ 9. ΜΟΡΦΗ ΠΑΚΕΤΟΥ ΑΠΟΣΤΟΛΗΣ ΔΕΔΟΜΕΝΩΝ ΑΠΟ LOGGER (SERVER)

Θέση byte	**Περιγραφή**	**Μήκος/Τύπος (Little Endian)**
0	Καρδιογράφημα (ECG)	8 (double)
8	Οξύμετρο (PPG IR LED)	8 (double)
16	Οξύμετρο (PPG RED LED)	8 (double)
24	Θερμοκρασία (Temp)	8 (double)
32	Κορεσμός οξυγόνου (SpO2)	8 (double)
40	Καρδιακός ρυθμός (HR)	8 (double)
48	Χρονοσήμανση (Timestamp)	8 (double)

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΙΚΟΝΑ 6. ΔΙΑΓΡΑΜΜΑ ΡΟΗΣ ΠΡΟΓΡΑΜΜΑΤΟΣ CLIENT ΧΡΗΣΤΗ

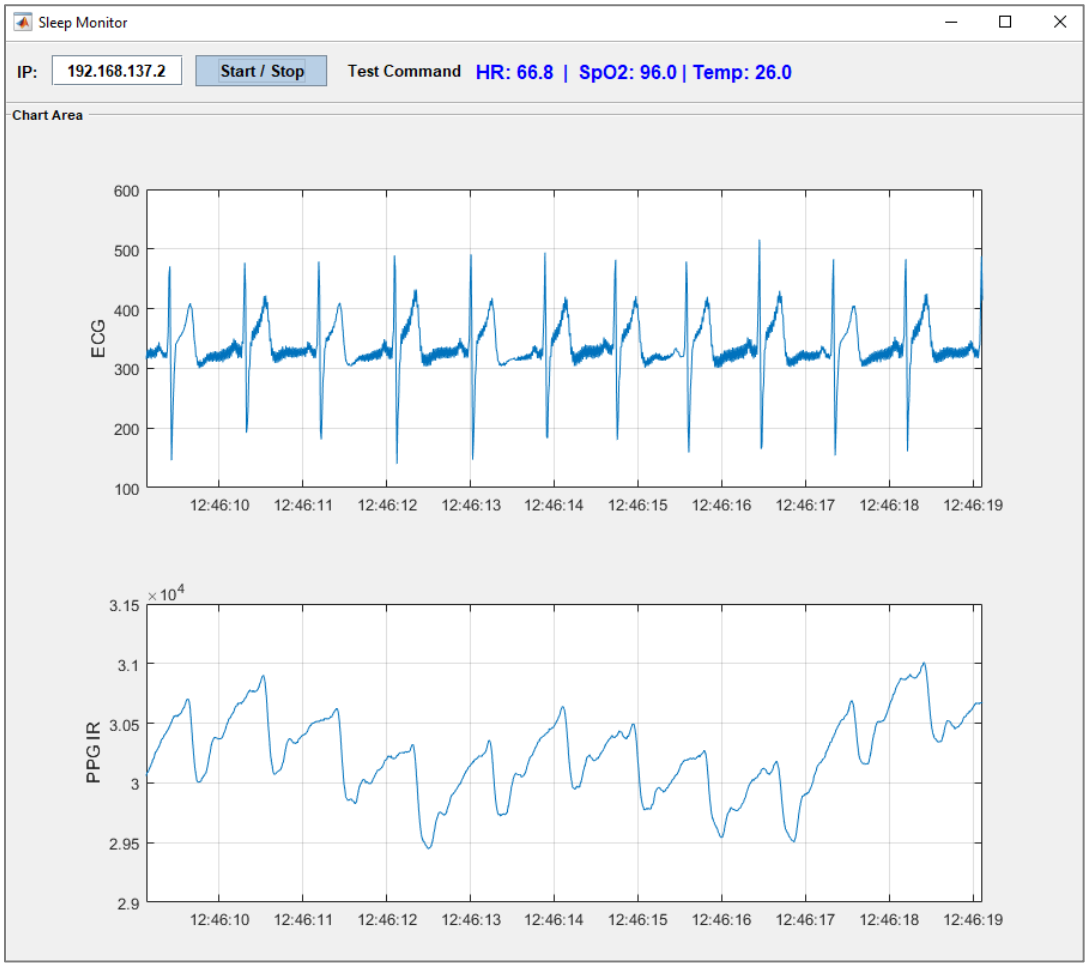
4.1. ΛΕΙΤΟΥΡΓΙΑ

Ο χρήστης εισάγει τη διεύθυνση IP του logger και ξεκινά ή σταματά την απεικόνιση σημάτων (Start/Stop). Η ανανέωση των εικόνων έχει τεθεί κάθε 3 sec, ενώ το παράθυρο έχει μήκος 10 sec. Κάθε 3 δευτερόλεπτα δηλαδή ξεκινά μια σύνδεση TCP/IP socket με το server (logger) και στέλνει αίτημα αποστολής δεδομένων ("data"). Τα σήματα που εμφανίζονται ως γραφικές παραστάσεις προέρχονται από καρδιογράφημα (ECG - Electrocardiogram) και από οξύμετρο (PPG - Photoplethysmogram), ενώ ως

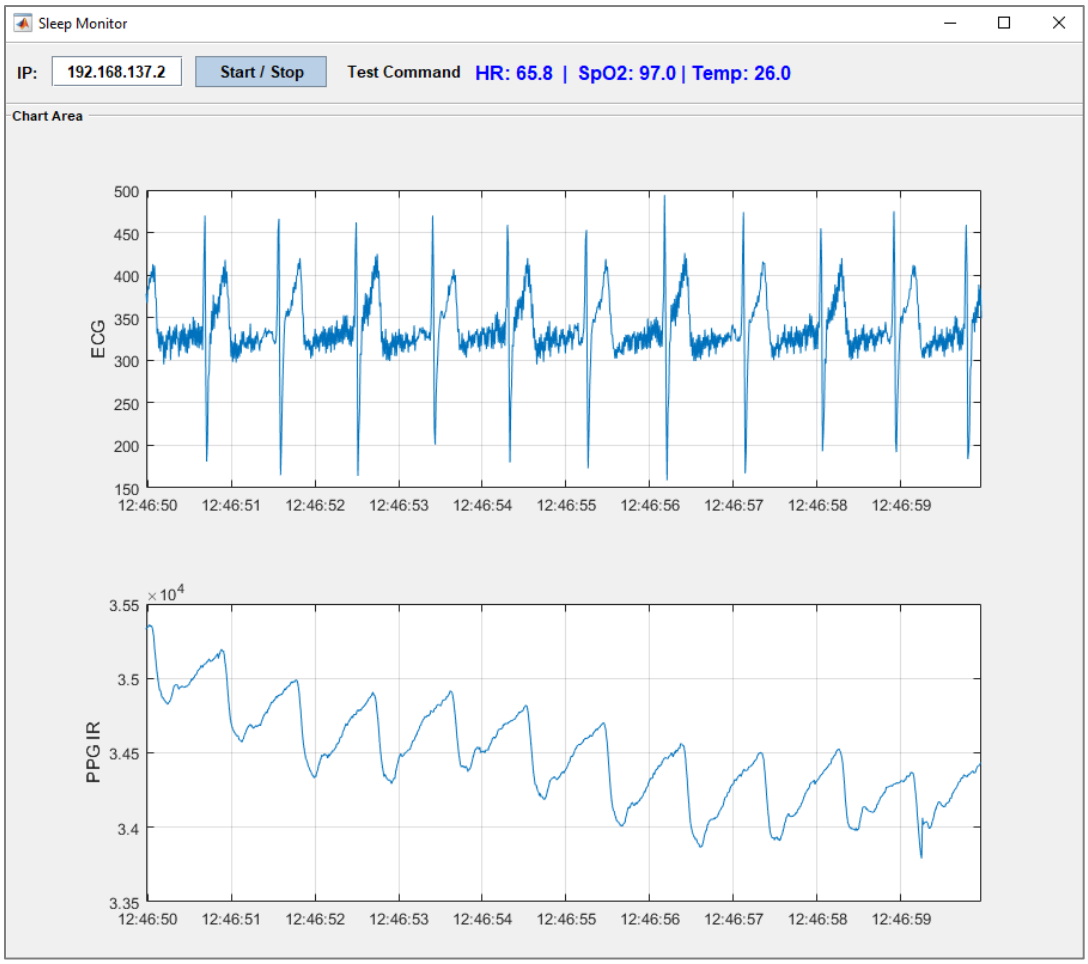
Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

μέσες τιμές ανά παράθυρο δίνονται ο καρδιακός ρυθμός (HR – Heart Rate), ο κορεσμός οξυγόνου (SpO2) και η θερμοκρασία. (*Η θερμοκρασία προέρχεται από τον ενσωματωμένο αισθητήρα του οξυμέτρου).* Ο χρήστης μπορεί να στέλνει εντολές στη μονάδα αισθητήρων μέσω του logger. Με το κουμπί [Test Command] δίνεται η demo εντολή ("cmd1") που αποκωδικοποιείται με την εναλλαγή κατάστασης On/Off στο πράσινο LED στο breadboard της πλακέτας sensor. Οι παρακάτω εικόνες είναι παραδείγματα που μπορεί να δει ο χρήστης.

Στιγμιότυπα παρακολούθησης:



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Βιβλιογραφία

Allen, J. (2007). Photoplethysmography and its application in clinical physiological measurement. *Physiological Measurement*, R1--R39.

Analog Devices. (2018). Ανάκτηση από Single-Lead, Heart Rate Monitor Front End: https://www.analog.com/media/en/technical-documentation/data-sheets/AD8232.pdf

Arduino. (2019). Ανάκτηση από Arduino Uno Rev3: https://store.arduino.cc/arduino-uno-rev3

DFRobot - Quality Arduino Robot IOT DIY Electronic Kit. (2019). Ανάκτηση από Gravity: Arduino Heart Rate Monitor Sensor (ECG) -DFRobot: https://www.dfrobot.com/product-1510.html

Electrónica Estudio - Ingeniería Electrónica y Proyectos PICmicro®. (2010). Ανάκτηση από HC-05 datasheet: http://www.electronicaestudio.com/docs/istd016A.pdf

MATLAB - MathWorks. (2019). Ανάκτηση από https://www.mathworks.com/products/matlab.html

Maxim Integrated - Analog, linear, & mixed-signal devices. (2014). Ανάκτηση από MAX30100: https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf

Microchip Technology. (2019). Ανάκτηση από ATmega328: https://www.microchip.com/wwwproducts/en/ATmega328

Peko's Library. (2018). Ανάκτηση από RCWL-0530/MAX30100 an oximetry / heart rate / temperature sensor: https://pekolibrary.wordpress.com/2018/05/27/rcwl-0530-max30100/

ProtoCentral HealthyPi v3 3.1.0. (2019). Ανάκτηση από ProtoCentral HealthyPi v3 3.1.0: http://healthypi.protocentral.com/

Raspberry Pi. (2018). Ανάκτηση από Raspberry Pi 3 Model B+: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

Raspberry Pi. (2019). Ανάκτηση από Download Raspbian for Raspberry Pi: https://www.raspberrypi.org/downloads/raspbian/

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Παράρτημα Ι

Λογισμικό μονάδας αισθητήρων

```
#include <Arduino.h>

#include "My_PulseOximeter.h"
using namespace Sleep;

PulseOximeter::PulseOximeter() :
    state(PULSEOXIMETER_STATE_INIT),
    tsFirstBeatDetected(0),
    tsLastBeatDetected(0),
    tsLastBiasCheck(0),
    tsLastCurrentAdjustment(0),
    redLedCurrentIndex((uint8_t)DEFAULT_LED_CURRENT),
    irLedCurrent(DEFAULT_LED_CURRENT),
    onBeatDetected(NULL)
{
}

bool PulseOximeter::begin(PulseOximeterDebuggingMode debuggingMode_)
{
    debuggingMode = debuggingMode_;

    bool ready = hrm.begin();

    if (!ready) {
        if (debuggingMode != PULSEOXIMETER_DEBUGGINGMODE_NONE) {
            Serial.println("Failed to initialize the HRM sensor");
        }
        return false;
    }

            sample_ready=false;

    hrm.setMode(MAX30100_MODE_SPO2_HR);
    hrm.setLedsCurrent(DEFAULT_LED_CURRENT, DEFAULT_LED_CURRENT);
    hrm.setLedsPulseWidth(MAX30100_SPC_PW_1600US_16BITS);
    hrm.setSamplingRate(MAX30100_SAMPRATE_100HZ);
    hrm.setHighresModeEnabled(true);

            hrm.startTemperatureSampling();

    irDCRemover = DCRemover(DC_REMOVER_ALPHA);
    redDCRemover = DCRemover(DC_REMOVER_ALPHA);

    state = PULSEOXIMETER_STATE_IDLE;

    return true;
}

void PulseOximeter::update()
{
    hrm.update();
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
    checkSample();
    checkCurrentBias();
}

float PulseOximeter::getHeartRate()
{
    return beatDetector.getRate();
}

uint8_t PulseOximeter::getSpO2()
{
    return spO2calculator.getSpO2();
}

uint8_t PulseOximeter::getRedLedCurrentBias()
{
    return redLedCurrentIndex;
}

void PulseOximeter::setOnBeatDetectedCallback(void (*cb)())
{
    onBeatDetected = cb;
}

void PulseOximeter::setIRLedCurrent(LEDCurrent irLedNewCurrent)
{
    irLedCurrent = irLedNewCurrent;
    hrm.setLedsCurrent(irLedCurrent, (LEDCurrent)redLedCurrentIndex);
}

void PulseOximeter::shutdown()
{
    hrm.shutdown();
}

void PulseOximeter::resume()
{
    hrm.resume();
}

void PulseOximeter::checkSample()
{
    // Dequeue all available samples, they're properly timed by the HRM
    while (hrm.getRawValues(&rawIRValue, &rawRedValue)) {

        temperature = hrm.retrieveTemperature();
                heartValue = analogRead(HEART_PIN);
                sample_ready=true;

      float irACValue = irDCRemover.step(rawIRValue);
      float redACValue = redDCRemover.step(rawRedValue);

      // The signal fed to the beat detector is mirrored since the cleanest monotonic spike is below
zero
      float filteredPulseValue = lpf.step(-irACValue);
      bool beatDetected = beatDetector.addSample(filteredPulseValue);

      if (beatDetector.getRate() > 0) {
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΠΑ & ΤΣ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
            state = PULSEOXIMETER_STATE_DETECTING;
            spO2calculator.update(irACValue, redACValue, beatDetected);
        } else if (state == PULSEOXIMETER_STATE_DETECTING) {
            state = PULSEOXIMETER_STATE_IDLE;
            spO2calculator.reset();
        }

        switch (debuggingMode) {
            case PULSEOXIMETER_DEBUGGINGMODE_RAW_VALUES:
                Serial.print("R:");
                Serial.print(rawIRValue);
                Serial.print(",");
                Serial.println(rawRedValue);
                break;

            case PULSEOXIMETER_DEBUGGINGMODE_AC_VALUES:
                Serial.print("R:");
                Serial.print(irACValue);
                Serial.print(",");
                Serial.println(redACValue);
                break;

            case PULSEOXIMETER_DEBUGGINGMODE_PULSEDETECT:
                Serial.print("R:");
                Serial.print(filteredPulseValue);
                Serial.print(",");
                Serial.println(beatDetector.getCurrentThreshold());
                break;

            default:
                break;
        }

        if (beatDetected && onBeatDetected) {
            onBeatDetected();
        }
    }
}

void PulseOximeter::checkCurrentBias()
{
    // Follower that adjusts the red led current in order to have comparable DC baselines between
    // red and IR leds. The numbers are really magic: the less possible to avoid oscillations
    if (millis() - tsLastBiasCheck > CURRENT_ADJUSTMENT_PERIOD_MS) {
        bool changed = false;
        if (irDCRemover.getDCW() - redDCRemover.getDCW() > 70000 && redLedCurrentIndex <
MAX30100_LED_CURR_50MA) {
            ++redLedCurrentIndex;
            changed = true;
        } else if (redDCRemover.getDCW() - irDCRemover.getDCW() > 70000 && redLedCurrentIndex >
0) {
            --redLedCurrentIndex;
            changed = true;
        }

        if (changed) {
            hrm.setLedsCurrent(irLedCurrent, (LEDCurrent)redLedCurrentIndex);
            tsLastCurrentAdjustment = millis();
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
            if (debuggingMode != PULSEOXIMETER_DEBUGGINGMODE_NONE) {
                Serial.print("I:");
                Serial.println(redLedCurrentIndex);
            }
        }

        tsLastBiasCheck = millis();
    }
}
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Παράρτημα ΙΙ

Λογισμικό μονάδας παρακολούθησης/καταγραφής

```
#include "WirelessSensor.h"
#include "DataServer.h"
#include "FrameProcess.h"
#include "MsgPrinter.h"
#include "LogManager.h"

#include <unistd.h>
#include <thread>
#include <iomanip>

using namespace std;

WirelessSensor *wwss;
DataServer *dss;
FrameProcess *fp;
LogManager *lm;


int main()
{

	PRINT("Sleep monitoring started!");

	// wireless sensor thread
	wwss = new WirelessSensor();
	thread t1(&WirelessSensor::start, wwss);

	// frame decoding thread
	fp = new FrameProcess(wwss);
	thread t2(&FrameProcess::start, fp);

	// network data server thread
	dss = new DataServer(wwss);
	thread t3(&DataServer::start, dss);

	// periodic save data to disk (usb) thread
	lm = new LogManager(wwss);
	thread t4(&LogManager::start, lm);

	// report every 10 sec
	while (1) {
		sleep(10);
		// print total incoming bytes
		PRINT("Received bytes: " + std::to_string(wwss->rec_bytes));
		// print total frame synchronization error attempts
		PRINT("Sync errors: " + std::to_string(wwss->sync_errors));
	}

	return 0;
}
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
#include "WirelessSensor.h"
#include "MsgPrinter.h"

#include <unistd.h>
#include <sys/socket.h>
#include <bluetooth/bluetooth.h>
#include <bluetooth/rfcomm.h>
#include <errno.h>

WirelessSensor::WirelessSensor()
{
	buffer.setSize(1024*1024);
	//fs = 100.0;	// nominal value for sampling rate
	fs = 99.58;		// calibrated sampling rate of sensor
	cmd = 0;
	log_bytes_saved = 0;
	rec_bytes = 0;
}


WirelessSensor::~WirelessSensor()
{
}


void WirelessSensor::start()
{
	// connect to Sleep Sensor bluetooth device
	// check for disconnections or no stream and reconnect
	try {

		struct sockaddr_rc addr = { 0 };
		int s, status;
		// sleep sensor bluetooth MAC (HC-05)
		char dest[] = { "00:18:E4:36:37:25" };
		char buf[1024] = { 0 };
		int bytes_read;
		struct timeval tv;
		tv.tv_sec = 2;
		tv.tv_usec = 0;
		s = socket(AF_BLUETOOTH, SOCK_STREAM, BTPROTO_RFCOMM);
		setsockopt(s, SOL_SOCKET, SO_RCVTIMEO, (const char*)&tv, sizeof(tv));
		addr.rc_family = AF_BLUETOOTH;
		addr.rc_channel = (uint8_t)1;
		str2ba(dest, &addr.rc_bdaddr);

		DEBUG("BT Connecting...");
		sleep(2);
		status = connect(s, (struct sockaddr *)&addr, sizeof(addr));
		if (status == 0) {
			DEBUG("BT connection OK!");
			struct timespec ts;
			time_t begin, end;
			clock_gettime(CLOCK_MONOTONIC_RAW, &ts);
			begin = ts.tv_sec;
			while (1) {
				bytes_read = read(s, buf, sizeof(buf));
				if (bytes_read > 0) {
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
                                        // incoming data from sensor
                                        if (buffer.writeAvailable() > bytes_read) {
                                                for (int j = 0; j < bytes_read; j++) {
                                                        bool ok = buffer.write(buf[j]);
                                                        if (ok == false) {
                                                                DEBUG("Buffer overrun!");
                                                                sleep(1);
                                                        }
                                                        else { rec_bytes++; }
                                                }
                                        }
                                        else {
                                                DEBUG("Buffer critical!");
                                        }

                                        clock_gettime(CLOCK_MONOTONIC_RAW, &ts);
                                        begin = ts.tv_sec;
                                        usleep(1000);
                                }
                                else {
                                        // check time out error
                                        clock_gettime(CLOCK_MONOTONIC_RAW, &ts);
                                        end = ts.tv_sec;
                                        double elapsed_secs = end - begin;
                                        if (elapsed_secs > 1) {
                                                DEBUG("BT connection timeout!");
                                                sleep(3);
                                                close(s);
                                                start();
                                        }
                                }

                                // send command code to sensor if there is one (non zero)
                                if (cmd != 0) {
                                        write(s, &cmd, sizeof(cmd));
                                        DEBUG("Command sent!");
                                        cmd = 0;
                                }

                                // check saved bytes and erase them from memory
                                if (log_bytes_saved > 0) {
                                        deleteElements(log_bytes_saved);
                                        log_bytes_saved = 0;
                                }
                        }
                }

                DEBUG("BT connection error! Code: ");
                DEBUG(errno);
                close(s);
                start();

        }
        catch (const exception &ex) {
                DEBUG("Sensor Exception:");
                DEBUG(ex.what());
        }
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΠΑ & ΤΣ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
}

void WirelessSensor::sendCmd(uint8_t c)
{
	cmd = c;
}

void WirelessSensor::deleteElements(int num)
{
	// delete sensor streaming elements from buffers after succesful write to disk
	std::vector<decltype(ecg)::value_type>(ecg.begin() + num, ecg.end()).swap(ecg);
	std::vector<decltype(ppg_ir)::value_type>(ppg_ir.begin() + num,
ppg_ir.end()).swap(ppg_ir);
	std::vector<decltype(ppg_red)::value_type>(ppg_red.begin() + num,
ppg_red.end()).swap(ppg_red);
	std::vector<decltype(temp)::value_type>(temp.begin() + num, temp.end()).swap(temp);
	std::vector<decltype(spo2)::value_type>(spo2.begin() + num, spo2.end()).swap(spo2);
	std::vector<decltype(heart_rate)::value_type>(heart_rate.begin() + num,
heart_rate.end()).swap(heart_rate);
	std::vector<decltype(time_vec)::value_type>(time_vec.begin() + num,
time_vec.end()).swap(time_vec);
}
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
#include "DataServer.h"
#include "MsgPrinter.h"

#include <cstring>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

using namespace std;

DataServer::DataServer(WirelessSensor *parent)
{
	sensor = parent;
}


DataServer::~DataServer()
{
}

void DataServer::start()
{
	// start TCP/IP server, waiting client requests (matlab program)

	try {
		int sockfd, newsockfd, portno = 11111;
		socklen_t clilen;
		const int buf_size = 1024;
		char buffer[1024];
		struct sockaddr_in serv_addr, cli_addr;
		int n;

		sockfd = socket(AF_INET, SOCK_STREAM, 0);
		if (sockfd < 0) {
			DEBUG("Error opening net socket!");
			return;
		}

		bzero((char *)&serv_addr, sizeof(serv_addr));
		serv_addr.sin_family = AF_INET;
		serv_addr.sin_addr.s_addr = INADDR_ANY;
		serv_addr.sin_port = htons((uint16_t)portno);
		if (bind(sockfd, (struct sockaddr *) &serv_addr, sizeof(serv_addr)) < 0) {
			DEBUG("ERROR on binding");
			return;
		}

		// Maximum size for queue = 5
		listen(sockfd, 5);
		clilen = sizeof(cli_addr);

		struct timeval tv;
		tv.tv_sec = 3;
		tv.tv_usec = 0;

		// server started
		while (1) {
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
                        DEBUG("Waiting for client...");
                        newsockfd = accept(sockfd, (struct sockaddr *) &cli_addr, &clilen);
                        setsockopt(newsockfd, SOL_SOCKET, SO_RCVTIMEO, (const char*)&tv,
sizeof tv);
                        if (newsockfd < 0) {
                                DEBUG("Error accepting connection!");
                                break;
                        }
                        else {
                                DEBUG("Client connected!");
                                struct timespec ts;
                                time_t begin, end;
                                clock_gettime(CLOCK_MONOTONIC_RAW, &ts);
                                begin = ts.tv_sec;
                                while (1) {
                                        n = read(newsockfd, buffer, buf_size);
                                        if (n < 0) {
                                                //no data
                                                clock_gettime(CLOCK_MONOTONIC_RAW, &ts);
                                                end = ts.tv_sec;
                                                time_t elapsed_secs = end - begin;
                                                if (elapsed_secs > 5) {
                                                        DEBUG("Server connection timeout!");
                                                        break;
                                                }
                                        }
                                        else {
                                                // get user request and response accordingly
                                                buffer[n] = '\0';
                                                string s(buffer);
                                                sendData(s, newsockfd);
                                                close(newsockfd);
                                                break;
                                        }
                                        usleep(10000);
                                }

                        }
                }

                close(newsockfd);
                close(sockfd);
                sleep(1);
                start();

        }
        catch (const exception &ex) {
                DEBUG("Server Exception:");
                DEBUG(ex.what());
        }

}

void DataServer::sendData(string cmd, int sockfd)
{

        char ok_resp[] = { "OK!" };
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ
2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
        // user request raw streaming data
        if (cmd.compare("data") == 0) {
                DEBUG("Sending data...");
                // send latest 10 sec of samples
                const float time_range_sec = 10.0;
                int len = sensor->ecg.size();
                int sample_count = (int)(time_range_sec * sensor->fs);
                const int fsz = sizeof(double);
                const int vec_num = 7;
                int byte_len = sample_count * fsz * vec_num;

                byte *bb = new byte[byte_len];
                int i = 0;
                double x;

                // get most recent samples stored in buffers
                for (int j = (len - sample_count); j < len; j++) {
                        if (j < 0) { continue; }

                        x = (double)sensor->ecg.at(j); memcpy(&bb[i], (byte*)&x, fsz);
                        x = (double)sensor->ppg_ir.at(j); memcpy(&bb[i + fsz], (byte*)&x, fsz);
                        x = (double)sensor->ppg_red.at(j); memcpy(&bb[i + 2 * fsz], (byte*)&x,
fsz);
                        x = (double)sensor->temp.at(j); memcpy(&bb[i + 3 * fsz], (byte*)&x, fsz);
                        x = (double)sensor->spo2.at(j); memcpy(&bb[i + 4 * fsz], (byte*)&x, fsz);
                        x = (double)sensor->heart_rate.at(j); memcpy(&bb[i + 5 * fsz], (byte*)&x,
fsz);
                        x = (double)sensor->time_vec.at(j); memcpy(&bb[i + 6 * fsz], (byte*)&x,
fsz);

                        i += vec_num * fsz;
                }
                write(sockfd, bb, byte_len);
                delete bb;
                DEBUG("OK!");
                return;
        }

        // user sent a command to the sensor
        if (cmd.compare("cmd1") == 0) {
                DEBUG("Send command to sensor");
                sensor->sendCmd(1);    // demo command code=1        (toggle sensor green LED)
                write(sockfd, ok_resp, strlen(ok_resp));
                return;
        }

        DEBUG("Unknown command!");
}
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
#include "FrameProcess.h"
#include "MsgPrinter.h"
#include <unistd.h>
#include <ctime>

FrameProcess::FrameProcess(WirelessSensor *parent)
{
	sensor = parent;
	sensor->buffer.setReadPos(0);
	prev_index = 0x1000000;
	last_index = 0;
}

FrameProcess::~FrameProcess()
{
}

// periodically get incoming bytes and decode them according to frame structure
void FrameProcess::start()
{
	const int framesize = 27;
	uint8_t frame[framesize];

	while (1)
	{
		if (sensor->buffer.readAvailable() < framesize)
		{
			usleep(1000);
			continue;
		}

		bool ok;
		for (int j = 0; j < framesize; j++) {
			frame[j] = sensor->buffer.read(ok);
			if (ok == false) {
				usleep(1000);
				start();
			}
		}

		uint8_t start1 = (uint8_t)frame[0];
		uint8_t start2 = (uint8_t)frame[1];
		uint8_t protoc = (uint8_t)frame[4];
		uint8_t foot = (uint8_t)frame[25];
		uint8_t end = (uint8_t)frame[26];

		// check frame start, end bytes
		if (start1 != 0x0a || start2 != 0xfa || protoc != 0x02 ||
			foot != 0x00 || end != 0x0b) {
			sensor->buffer.moveReadPos(1 - framesize);
			// increase synchronization error count, move to next byte and try again
			sensor->sync_errors++;
			continue;
		}
		// next frame

		// get the sample inside frame and store to corresponding lists
		sensor->ecg.push_back((float)*(int16_t*)&frame[5]);
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
                sensor->ppg_ir.push_back((float)*(int32_t*)&frame[9]);
                sensor->ppg_red.push_back((float)*(int32_t*)&frame[13]);
                sensor->temp.push_back((float)*(int16_t*)&frame[17]);
                sensor->spo2.push_back(*(uint8_t*)&frame[20]);
                sensor->heart_rate.push_back(*(uint8_t*)&frame[21]);

                // the next 3 bytes is the frame serial number (index)
                // it will be translated to timestamp
                uint8_t a = *(uint8_t*)&frame[22];
                uint8_t b = *(uint8_t*)&frame[23];
                uint8_t c = *(uint8_t*)&frame[24];
                double new_index = (c * 256 + b) * 256 + a;

                // time syncing with incoming samples
                if (new_index <= prev_index)
                {
                        prev_time = time(nullptr);
                        prev_index = new_index;
                        last_index = new_index - 1;
                }

                double di = new_index - last_index;
                if (di != 1) {
                        // lost frames if index not increased by 1
                        DEBUG("Frames lost: "+ std::to_string((int)di-1));
                }

                double cur_ms = (1000.0/sensor->fs)*(new_index - prev_index);
                double msec = ((double)prev_time) * 1000 + cur_ms;
                sensor->time_vec.push_back(msec);

                last_index = new_index;
        }
}
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
#include "MsgPrinter.h"

#include <iostream>

using namespace std;

MsgPrinter::MsgPrinter()
{
}

MsgPrinter::~MsgPrinter()
{
}

// used for runtime printing messages
void MsgPrinter::print()
{
	cout << str() << endl;
	str("");
}

// used for debugging messages
void MsgPrinter::debugout()
{
	// comment out next line to disable DEBUG out
	print();

}
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
#include "LogManager.h"
#include "MsgPrinter.h"
#include <unistd.h>
#include <iostream>
#include <fstream>
#include <cerrno>
#include <cstring>

LogManager::LogManager(WirelessSensor *parent)
{
	sensor = parent;
	log_time = 60;  // 60 sec period for write to disk
}


LogManager::~LogManager()
{

}

void LogManager::start()
{
	// periodically check available buffer samples and save them to usb disk
	while (1) {

		if (sensor->log_bytes_saved == 0) {
			// check available vector size
			int realtime_bytes = (int)sensor->fs * 60;		// 1min for real time view
			// we do not save the latest 1 min (preserved for preview)
			int min_byte_num = (int)sensor->fs * log_time + realtime_bytes;
			if ((int)sensor->ecg.size() > min_byte_num) {
				// save samples to disk
				int byte_num = sensor->ecg.size() - realtime_bytes;
				// save oldest N=byte_num samples from buffer
				bool ok = saveToFile(byte_num);
				if (ok) { sensor->log_bytes_saved = byte_num; }
				else {
					// could not save to disk
				}
			}
		}
		// repeat every 5 sec
		sleep(5);
	}

}

// save to usb, return true if succesful
bool LogManager::saveToFile(int num)
{
	ofstream ofs;
	int usbdir;
for (usbdir = 0; usbdir <= 3; usbdir++) {
	string fname = "/media/usb" + std::to_string(usbdir) + "/data.bin";
	ofs.open(fname, ofstream::binary | ofstream::app);
	if (ofs.is_open()) {
		DEBUG("Writing to disk " + std::to_string(usbdir));
		for (int j = 0; j < num; j++) {
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
                        ofs.write((char*)&sensor->ecg.at(j),4);
                        ofs.write((char*)&sensor->ppg_ir.at(j),4);
                        ofs.write((char*)&sensor->temp.at(j),4);
                        ofs.write((char*)&sensor->spo2.at(j),1);
                        ofs.write((char*)&sensor->heart_rate.at(j),1);
                        ofs.write((char*)&sensor->time_vec.at(j),8);
                        // 22 bytes to save
                }
                ofs.close();
                DEBUG(std::to_string(num) + " samples saved!");
                return true;
        }
        else {
                // error on opening file
                DEBUG(std::strerror(errno));
                continue;          // next mountpoint
        }
}
        DEBUG("Write disk error!");
        return false;
}
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Παράρτημα ΙΙΙ

Λογισμικό παρακολούθησης

```
function varargout = SleepMonitorM(varargin)


gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @SleepMonitorM_OpeningFcn, ...
                   'gui_OutputFcn',  @SleepMonitorM_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end


if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end



% --- Executes just before SleepMonitorM is made visible.
function SleepMonitorM_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to SleepMonitorM (see VARARGIN)


% Choose default command line output for SleepMonitorM
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
h = findobj(0, 'tag', 'figure1');
p = get(h, 'position');
p(3) = 800;
p(4) = 600;
set(h, 'position', p);
movegui(h,'center');
obj = findobj(0, 'tag', 'valuestag');
set(obj, 'string', '');

disp('Starting Sleep Monitor...');

% UIWAIT makes SleepMonitorM wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = SleepMonitorM_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function iptag_Callback(hObject, eventdata, handles)
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
% hObject    handle to iptag (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of iptag as text
%        str2double(get(hObject,'String')) returns contents of iptag as a double


% --- Executes during object creation, after setting all properties.
function iptag_CreateFcn(hObject, eventdata, handles)
% hObject    handle to iptag (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in plottag.
function plottag_Callback(hObject, eventdata, handles)
% hObject    handle to plottag (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% get Start/Stop button state
onoff=get(handles.plottag, 'Value');
while onoff
    ip = get(handles.iptag, 'String');
    y = getData('data', ip);
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
    if ~isempty(y)
try
        plotData(y);
catch ME
    disp([ME.identifier, ', ', ME.message]);
    return;
end
    end
    pause(3);
    onoff=get(handles.plottag, 'Value');
end

% --- Executes on button press in cmd1tag.
function cmd1tag_Callback(hObject, eventdata, handles)
% hObject    handle to cmd1tag (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
ip = get(handles.iptag, 'String');
y = getData('cmd1', ip);

% --- Executes when figure1 is resized.
function figure1_SizeChangedFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

obj = findobj(0, 'tag', 'figure1');
set(obj, 'units', 'points');
p=get(obj, 'position');
a = 40;
y = p(4)-a;
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
obj = findobj(0, 'tag', 'inputpanel');
set(obj, 'position', [0, y, p(3), a]);

obj = findobj(0, 'tag', 'plotpanel');
set(obj, 'position', [0, 0, p(3), y]);



function plotData(x)

n = length(x);
bpn = 8;   %float precision 8bytes per number
npf = 7;    % numbers per frame
N = floor(n/(npf*bpn))*npf*bpn; % total bytes
fnums = typecast(x(1:N),'double');   % convert to floats
N = length(fnums);
A = reshape(fnums, npf, N/npf).';

ecg = A(:,1);        % ecg channel
ppg_ir = A(:,2);     % oximeter IR channel
temp = A(:,4);       % temperature

sp02 = A(:,5);       % oxygen saturation
hr = A(:,6);         % heart rate (bpm)

mspo2 = mean(sp02);    % mean SpO2 value
mhr = mean(hr);      % mean Heart Rate (bpm)
mtmp = mean(temp);      % mean temperature (C)

tdif=hours(tzoffset(datetime('now','TimeZone','Europe/Athens')));
t = (A(:,7)+tdif*3600*1000)/1000/86400 + datenum(1970,1,1,0,0,0);

pn = 2;     % total plot num
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
subplot(pn,1,1,'Parent',findobj(0, 'tag', 'plotpanel'));
plot(t, ecg);
grid on;
datetick('x');
xlim([t(1), t(end)]);
ylabel('ECG');

subplot(pn,1,2,'Parent',findobj(0, 'tag', 'plotpanel'));
plot(t, ppg_ir);
grid on;
datetick('x');
xlim([t(1), t(end)]);
ylabel('PPG IR');

obj = findobj(0, 'tag', 'valuestag');
ss = sprintf('HR: %.1f  |  SpO2: %.1f | Temp: %.1f',...
            mhr, mspo2, mtmp);
set(obj, 'string', ss);

end


clc;
clear;

% Sample code for opening Sleep Monitor data file stored by the logger

file_name = 'data.bin';

a = dir(file_name);
sz = a.bytes;
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
% sample size in bytes : 22
bps = 22;
x = mod(sz,bps);
if x~=0
    disp(x);
    disp('File data alignment is wrong!');
    return;
end

fid = fopen(file_name, 'r');
N=int32(sz/bps);
% prealocating space
ecg=single(zeros(N,1));
ppg=single(zeros(N,1));
temp=single(zeros(N,1));
spo2=uint8(zeros(N,1));
hr=uint8(zeros(N,1));
tv=double(zeros(N,1));
disp('Reading file...');
i=int32(0);
j=int32(N/10);
fprintf('Progress: 00%%');

X = fread(fid, inf, 'uint8');
fclose(fid);

for i=1:N
    if mod(i, j)==0; fprintf('\b\b\b%d%%',int32(i*100/N)); end;
    S = X((i-1)*bps+1:i*bps);
    ecg(i)=typecast(uint8(S(1:4)),'single');
    ppg(i)=typecast(uint8(S(5:8)),'single');
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
    temp(i)=typecast(uint8(S(9:12)),'single');
    spo2(i)=typecast(uint8(S(13:13)),'uint8');
    hr(i)=typecast(uint8(S(14:14)),'uint8');
    tv(i)=typecast(uint8(S(15:22)),'double');
end
fprintf('\b\b\b100%%\n');
fprintf('Done\n');

tdif=hours(tzoffset(datetime('now','TimeZone','Europe/Athens')));
ut = (tv+tdif*3600*1000)/86400.0/1000.0;
t = datenum(1970,1,1,0,0,0) + ut;

% plot stream
figure(1);
% plot(t, ecg);
plot(t, ppg);
grid on;
datetick('x');



function y = getData(x, ip)
y=[];

try
    t = tcpclient(ip, 11111 ,'Timeout', 5);
    write(t, uint8(x));
tic
while t.BytesAvailable<=0
    pause(0.1);
    if toc>5
        printMsg('Read timeout 1!');
        clear t;
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ
2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

```
        return;
    end
end

while t.BytesAvailable>0
    y=[y,read(t)];
end

catch ME
    disp([ME.identifier, ', ', ME.message]);
end

clear t;

end
```


Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ
ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΠΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Οπισθόφυλλο

«Η εργασία 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:Τ1ΕΔΚ-01958)»

This research has been co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH – CREATE – INNOVATE (project code:T1EDK-01958)

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΟΙΚΟΝΟΜΙΑΣ & ΑΝΑΠΤΥΞΗΣ
ΕΙΔΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΤΠΑ & ΤΣ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΕΠΑνΕΚ
ΕΠΑνΕΚ 2014-2020
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ
ΕΠΙΧΕΙΡΗΜΑΤΙΚΟΤΗΤΑ
ΚΑΙΝΟΤΟΜΙΑ
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης