



(12) CERERE DE BREVET DE INVENȚIE

(21) Nr. cerere: a 2014 00449

(22) Data de depozit: 16/06/2014

(41) Data publicării cererii:
30/03/2016 BOPI nr. 3/2016

(71) Solicitant:
• UNIVERSITATEA "ȘTEFAN CEL MARE"
DIN SUCEAVA, STR. UNIVERSITĂȚII NR. 13,
SUCEAVA, SV, RO

(72) Inventatori:
• CIUFUDEAN CĂLIN HORĂȚIU,
STR. ȘTEFAN CEL MARE NR. 4, BL. 6,
SC. A, AP. 4, VATRA DORNEI, SV, RO;

• BUZDUGA CORNELIU, STR. PUTNEI
NR. 520, VICOVU DE SUS, SV, RO;
• CIOCAN SORIN-ANDREI, SAT BURLEȘTI,
COMUNA UNȚENI, BT, RO;
• MIHĂILĂ ALEXANDRU-PETRU,
SAT CAPU CÂMPULUI, COMUNA CAPU
CÂMPULUI, SV, RO

(54) SISTEM DE MONITORIZARE A POZIȚIEI CORECTE LA
CALCULATOR

(57) Rezumat:

Invenția se referă la un sistem de monitorizare a poziției corecte a unui utilizator în fața calculatorului. Sistemul conform invenției este constituit dintr-un modul (1) care conține un microcontroler (M1), un accelerometru (A), un bloc de comunicație bluetooth (Bth) și o baterie (B), dintr-un modul (2) care conține niște senzori (SU) ultrasonici, și dintr-un modul (3) auxiliar, care conține niște LED-uri și niște difuzoare (D), modulele (1, 2, 3) fiind interconectate printr-un microcontroler (M2) ce are rol de luare a deciziilor privind informațiile provenite de la senzori (SU), care vor determina distanța corpului utilizatorului față de monitor, și poziția corectă a ochilor, asigurând în același timp și comunicația întregului sistem cu calculatorul (PC).

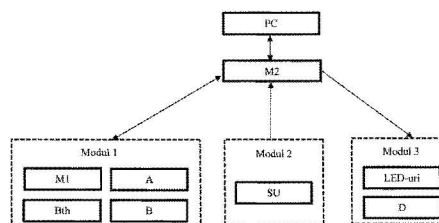
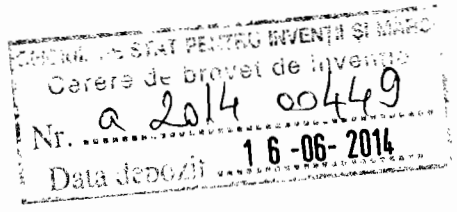


Fig. 1

Revendicări: 1
Figuri: 2

Cu începere de la data publicării cererii de brevet, cererea asigură, în mod provizoriu, solicitantului, protecția conferită potrivit dispozițiilor art.32 din Legea nr.64/1991, cu excepția cazurilor în care cererea de brevet de invenție a fost respinsă, retrasă sau considerată ca fiind retrasă. Întinderea protecției conferite de cererea de brevet de invenție este determinată de revendicările conținute în cererea publicată în conformitate cu art.23 alin.(1) - (3).





Sistem de monitorizare a poziției corecte la calculator

Invenția se referă la un sistem de monitorizare a poziției corecte a persoanelor care lucrează timp îndelungat în fața calculatorului.

În acest scop actualmente sunt cunoscute trei soluții: (1. LUMObac, <http://www.lumobodytech.com/lumoback/>, dispozitiv electronic folosit pentru atenționarea în timp real a operatorului uman atunci când acesta menține o poziție incorectă pe scaun sau la calculator. Se prezintă sub forma unei curele care se fixează pe zona lombară a spatelui. 2. RISR, <http://www.risr.me/>, dispozitiv format dintr-o rețea de senzori dispuși pe întregul trunchi al corpului utilizatorului. De asemenea, folosește o aplicație smartphone pentru monitorizarea poziției corpului. 3. PostureNOW, <http://www.posturenow.com/>, dispozitiv similar celor descrise mai sus, prezentat sub forma unei benzi elastice pe care este fixat un senzor și un microcontroler.

Aceste sisteme prezintă următoarele dezavantaje:

- vizează doar poziția spatelui și nu dispun de un sistem de monitorizare a poziției ochilor față de monitorul unui calculator;
- preț de cost ridicat;
- fiabilizate scăzută;
- utilizare greoaie.

Sistemul de monitorizare a poziției corecte la calculator, conform invenției, înlătură dezavantajele menționate mai sus, prin aceea că folosește tehnici de siguranță ce monitorizează atât poziția spatelui cât și poziția ochilor față de monitorul unui calculator.

Avantajele invenției sunt:

- asigură o monitorizare completă și eficientă;
- preț de cost redus;
- fiabilitate ridicată;
- utilizare simplă.

Se dă în continuare un exemplu de sistem de monitorizarea a poziției la calculator în legătură cu

- fig. 1 care reprezintă schema bloc a sistemului;
- fig. 2 care reprezintă schema logică de funcționare a sistemului.

Acest sistem asigură o poziție corectă la calculator a operatorului uman. Putem defini poziția corpului utilizatorului ca fiind corectă dacă se îndeplinesc o serie de condiții:

- ochii utilizatorului trebuie să se afle la o distanță cuprinsă între 50 cm și 70 cm față de monitor, iar privirea trebuie să fie orientată central pe suprafața monitorului;
- spatele utilizatorului trebuie menținut drept, însă acesta nu trebuie să se sprijine de spătarul scaunului.

Pentru a facilita îndeplinirea condițiilor amintite mai sus în ceea ce privește poziția utilizatorului la calculator, dispozitivul este constituit din două module principale:

- modulul 1, pentru asigurarea poziției corecte a spatelui și conține un microcontroller M1, un accelerometru A, un bloc de comunicație bluetooth Bth și o baterie B. Accelerometrul A va fi poziționat pe spatele utilizatorului cu ajutorul unui sistem de prindere și va furniza poziția spatelui în funcție de axele x, y, z. Microcontrolerul M1 va analiza datele primite de la accelerometru A și împreună cu modulul Bluetooth Bth datele vor fi trimise către microcontrolerul M2. Modulul 1 va fi alimentat prin bateria B.

- modulul 2, pentru asigurarea poziției corecte a ochilor față de monitor și a distanței optime față de acesta ce conține senzori ultrasonici SU ce vor determina distanța corpului utilizatorului față de monitor și poziția corectă a ochilor;

- modulul 3, este un modul auxiliar ce conține LED-uri și difuzoare D care în funcție de datele primite de la cele două module principale, va semnaliza dacă utilizatorul se află la o distanță regulamentară față de monitor sau dacă spatele acestuia se află într-o poziție corectă.

Aceste module sunt interconectate între ele printr-un microcontroler M2, care va

asigura și comunicația sistemului cu calculatorul PC. Totodată microcontrolerul M2 are rol în luarea deciziilor privind informațiile provenite de la senzorii ultrasonici SU.

Sistemul de monitorizare a poziției la calculator, conform invenției, poate fi reprodus cu aceleași caracteristici și performanțe ori de câte ori este necesar fapt care constituie un argument în vederea respectării criteriului de aplicabilitate industrială.

În continuare se dă o secvență de cod pentru achiziția datelor cu ajutorul senzorilor ultrasonici.

```
//Program pentru punerea in functiune //a doi senzori ultrasonici si a unui accelerometru //Aduugarea
//librării SPI necesara pentru comunicatia cu senzorul ADXL345 #include
<SPI.h> //Atribuirea semnalului Chip Select pinului 10.
int CS=10; //Configurarea regitrilor disponibili pentru ADXL345 char
OWER_CTL = 0x2D; //Power Control Register char
DATA_FORMAT = 0x31;char DATA0 = 0x32;
//X-Axis Data 0 char DATA1 = 0x33;
//X-Axis Data 1 char DATAY0 = 0x34;
//Y-Axis Data 0 char DATAY1 = 0x35;
//Y-Axis Data 1 char DATAZ0 = 0x36;
//Z-Axis Data 0 char DATAZ1 = 0x37;
//Z-Axis Data 1
//Acest buffer va stoca valorile citite de la regitrii senzorului ADXL345 char values[10];
//Variabile vor fi folosite pentru a stoca cele 3 axe ale accelerometrului
int x,y,z;
//Definire senzor ultrasunete1 + LED1 #define trigPin1 3 #define echoPin1 2 #define LED1 6
//Definire senzor ultrasunete2 + LED2 #define trigPin2 5 #define echoPin2 4 #define LED2 7
void setup() {
//Se initiaza comunicatia SPI
SPI.begin();
//Se configureaza conexiunea SPI pentru ADXL345
SPI.setDataMode(SPI_MODE3);
//Se creeaza o conexiune seriala pentru a afisa datele la terminal Serial.begin(9600);
writeRegister(DATA_FORMAT, 0x01);
writeRegister(POWER_CTL, 0x08);
//Setarea pinilor pinMode(CS, OUTPUT);
digitalWrite(CS, HIGH);
pinMode(trigPin1, OUTPUT);
pinMode(echoPin1, INPUT);
pinMode(LED1, OUTPUT);
pinMode(trigPin2, OUTPUT);
pinMode(echoPin2, INPUT);
pinMode(LED2, OUTPUT);
Serial.print(" Senzori \tAccelerometru"); }
void loop(){ int duration1, distance1, duration2, distance2;
//Citirea celor 6 bytes de date incepand de la regitruul DATA0 vor prelua
//cele 3 axe x, y si z de la ADXL345
//Rezultatele operatiei vor fi stocate in bufferul values[] readRegister(DATA0, 6, values);
//ADXL345 ofera valori de accelerare de 10-bi?i, dar ele sunt stocate ca
//bytes (8-biti). Pentru a ob?ine valoarea totala, doi octeti trebuie sa fie
```

```
//combinati pentru fiecare axã
//valoarea lui X este stocata in values[0] si values[1] x = ((int)values[1]<<8)|(int)values[0];
//Valoarea lui Y este stocata in values[2] si value[3] y = ((int)values[3]<<8)|(int)values[2];
//Valoarea lui Z este stocata in values[4] si values[5] z = ((int)values[5]<<8)|(int)values[4];
//Punerea in functiune a primului senzor ultrasonic digitalWrite(trigPin1, HIGH); delay(1);
digitalWrite(trigPin1, LOW); duration1 = pulseIn(echoPin1, HIGH); distance1 = (duration1/2) / 29.1;
//Punerea in functiune a celui de-al doilea senzor ultrasonic digitalWrite(trigPin2, HIGH); delay(1);
digitalWrite(trigPin2, LOW); duration2 = pulseIn(echoPin2, HIGH);
distance2 = (duration2/2) / 29.1;
//Rezultatele vor fi afisate la terminal
//Distanta senzor1 Serial.print("\n");
Serial.print(distance1); Serial.print(" cm");
//Distanta senzor2 Serial.print("\t");
Serial.print(distance1); Serial.print(" cm");
//Valori accelerometru Serial.print("\t");
Serial.print(x, DEC); Serial.print(',');
Serial.print(y, DEC); Serial.print(',');
Serial.print(z, DEC);
//Daca distanta furnizata de cei doi senzori este mai mare decat 60 cm
//atunci se aprinde LED1 -> distanta corecta fata de monitor if (distance1 && distance2 > 60)
digitalWrite(LED1, HIGH);
else digitalWrite(LED1, LOW);
//Daca distanta furnizata de cei doi senzori este mai mica decat 60 cm
//atunci se aprinde LED2 -> utilizatorul se afla la o distanta incorecta fata
//de monitor if (distance1 && distance2 < 60) digitalWrite(LED2, HIGH);
else digitalWrite(LED2, LOW);
delay(1000); } //In continuare sunt definite 2 functii:
//writeRegister() - va scrie o valoare catre un registru al ADXL345
//readRegister() - se vor citi un anumit numar de registri incepand cu o
//anumita adresa, registri ce vor fi memorati intr-un buffer
//Parametri pentru writeRegister:
//char registerAddress - registrul la care se va scrie valoarea
//char value - valoarea ce va fi scrisa la un anumit registru void writeRegister(char registerAddress, char
value){
//Se seteaza pinul Chip Select pe low pentru a semnala inceputul unui
//pachet SPI digitalWrite(CS, LOW);
//Adresa registrului este transferata catre SPI SPI.transfer(registerAddress);
//Se transfera valoarea dorita a unui registru catre SPI SPI.transfer(value);
//Se seteaza pinul Chip Select pe high pentru a semnala sfarsitul unui
//pachet SPI digitalWrite(CS, HIGH); }
//Parametri pentru readRegister:
//char registerAddress - registrul de adresa incepe citirea secventei
//int numBytes - numarul de registri ce vor fi cititi
//char *values - un pointer catre un buffer unde rezultatele vor fi memorate void readRegister(char
registerAddress, int numBytes, char * values){
//Se seteaza cel mai semnificativ bit al registrului de adresa char address = 0x80 | registerAddress;
//Daca se face o citire multi-byte, bitul 6 trebuie setat si el if(numBytes > 1)address = address | 0x40;
//Se seteaza pinul Chip Select pe low pentru a incepe un pachet SPI digitalWrite(CS, LOW);
//Se transfera registrul de adresa ce trebuie citit SPI.transfer(address);
//Citirea registrilor continua pana cand s-a citit numarul specificat
//Rezultatele vor fi stocate intr-un buffer for(int i=0; i<numBytes; i++){ values[i] = SPI.transfer(0x00);
}
//Se seteaza pinul Chip Select pe high pentru a sfarsi pachetul SPI digitalWrite(CS, HIGH); }
```

Sistemul de monitorizare a poziției corecte la calculator, conform invenției, poate fi reprodus cu aceleași caracteristici și performanțe ori de câte ori este necesar fapt care constituie un argument în vederea respectării criteriului de aplicabilitate industrială.

Revendicare

Sistem de monitorizare a poziției corecte la calculator, conform invenției este constituit din trei module: modulul 1 - conține un microcontroler (M1), un accelerometru (A), un bloc de comunicație bluetooth (Bth) și o baterie (B); modulul 2 - conține senzorii ultrasonici (SU); modulul 3 - este un modul auxiliar ce conține LED-uri și difuzoare (D), caracterizat prin aceea că, cele trei module sunt interconectate între ele printr-un microcontroler (M2), care are rol în luarea deciziilor privind informațiile provenite de la senzorii ultrasonici (SU) ce vor determina distanța corpului utilizatorului față de monitor și poziția corectă a ochilor, asigurând în același timp și comunicația întregului sistem cu calculatorul (PC).

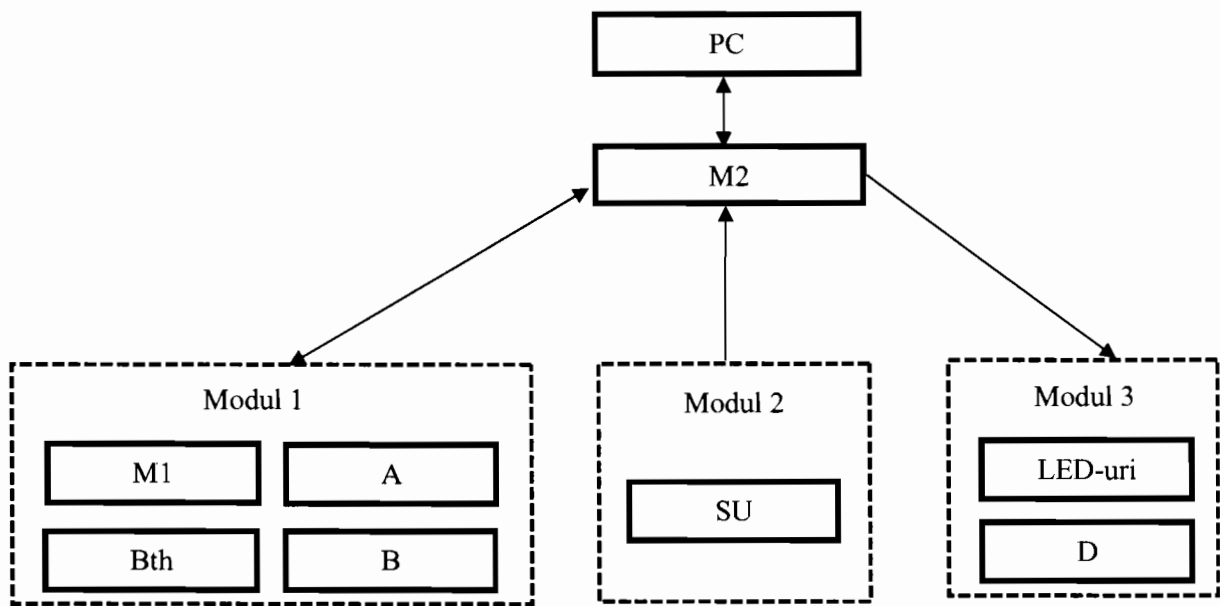


Fig. 1

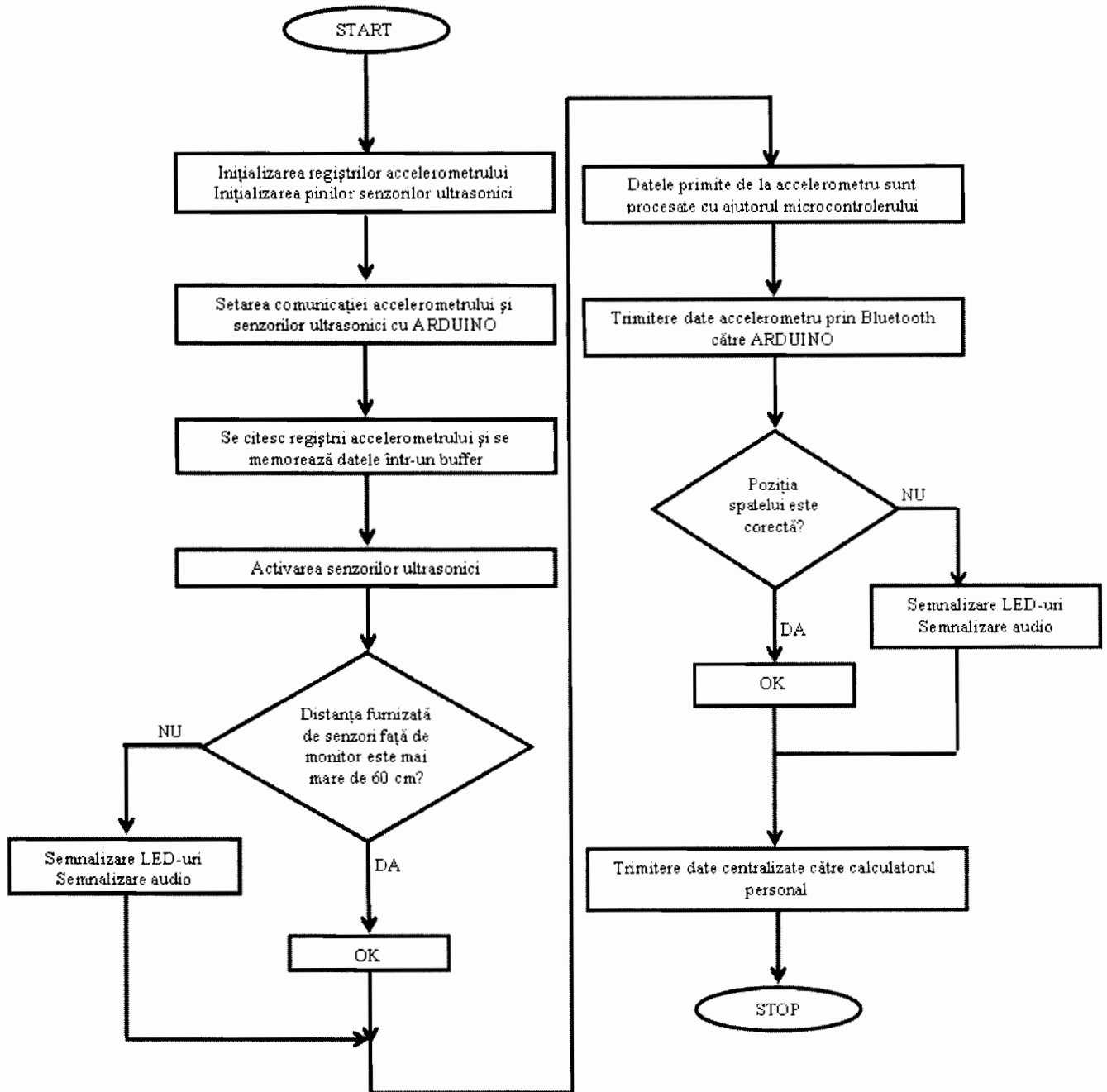


Fig. 2