Proiect Arduino

programare orientată pe obiecte

*Sistem ECG*

Studenți:

Bălănescu Adrian-Gabriel,

Bîzdoc Vasile-Gabriel,

Chirap Andrei,

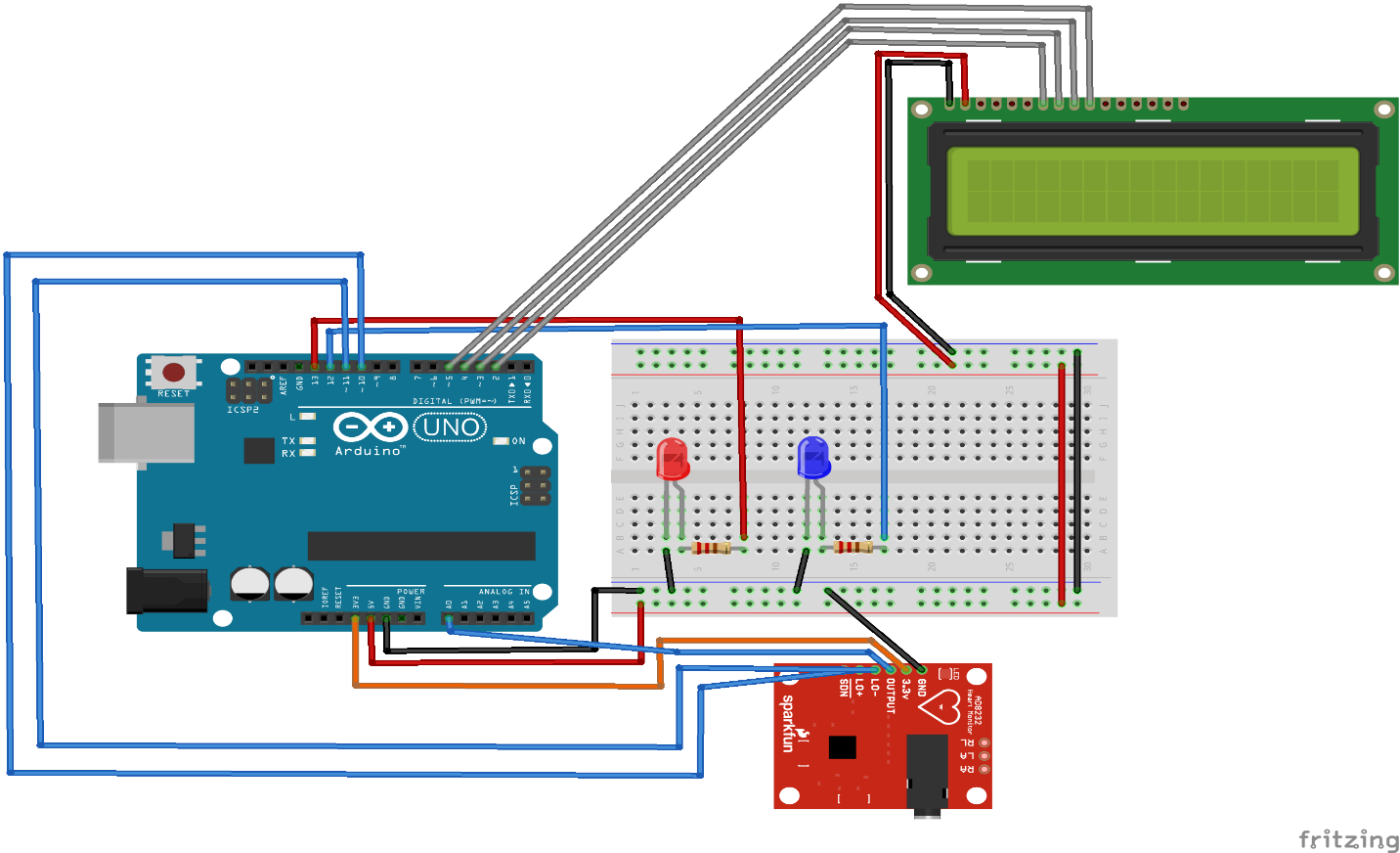
An II, Grupa 2.1

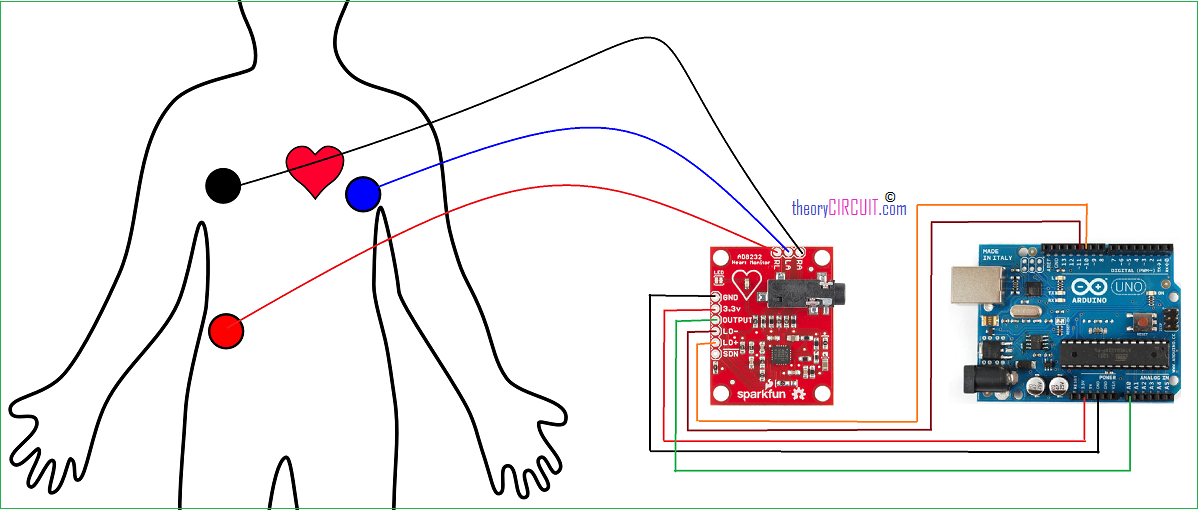
Timișoara, 2017

Cerințe inițiale:

1. Dacă sistemul este oprit se va aprinde ledul roșu, iar dacă este pornit ledul albastru. De asemenea se va afișa pe LCD un mesaj corespunzător.
2. Pe calculator, cu ajutorul programului *Processing* se va putea vizualiza semnalul de ECG, iar pe Serial Monitor se vor afișa valorile.

Diagramă hardware:





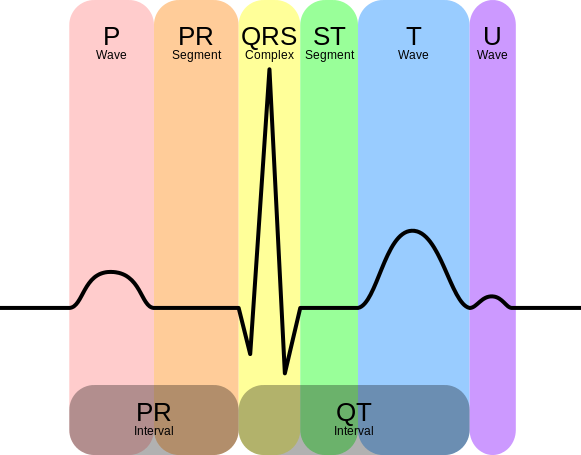
Componente sistem:

1. Placă de dezvoltare Arduino UNO v3. X1
2. Senzor ECG Sparkfun AD8232. X1
3. Led brick albastru. X1
4. Led brick roșu. X1
5. Shield LCD1602. X1

Obiectiv și mod de funcționare:

Acest proiect are ca scop înregistrarea și vizualizarea în timp real al unui semnal ECG (electrocardiogramă).

Semnalul ECG este separat în două intervale de bază – PR – și -QT- reprezentate mai jos:



Semnalul este înregistrat sub formă de impulsuri electromagnetice, stabilizate și amplificate de senzorul ECG și este vizualizat prin intermediul unui program în Processing care preia datele de la ieșirea Serială a plăcii Arduino.

Sistemul funcționează astfel:

* La detectarea unui semnal ECG sistemul trece în starea Activ, se aprinde ledul albastru, se afișează pe LCD un mesaj și începe înregistrarea și transmiterea semnalului la interfața serială. (este detectat un semnal ECG atunci cand semnalele de la pinii LO+ și LO- sunt 0, adică atunci când nu există interferențe în semnal sau electrozi deconectați)
* În Processing semnalul este preluat sub forma unui șir de valori de amplitudine care sunt desenate cu ajutorul unor linii.

În figura de mai jos este reprezentată schema UML a softwareului ce preia datele de la senzorul ECG.

*componentaSistem*

|  |
| --- |
| -pin |
| + componentaSistem()  + getPin()  + setOn()  + setOff() |

*Led*

|  |
| --- |
| -ledState |
| + led()  + setOn()  + setOff() |

*ecgSensor*

|  |
| --- |
| -state  -loPlus  -loMinus  -value  -loPlusPin  -loMinusPin |
| + ecgSensor()  + readData()  + getState() |

Din schema UML se observă clasa de bază *componentaSistem* care moștenește celelalte clase și conține funcții virtuale redefinite în celelalte clase.

Clasa *led* conține metode pentru pornirea și oprirea unui led.

Clasa *ecgSensor* conține metode pentru configurarea interfeței senzorului.

Schema logică:

//TODO

Testarea aplicației:

|  |  |  |
| --- | --- | --- |
| *Acțiune* | *Răspuns așteptat* | *Răspuns real* |
| * conectarea electrozilor în punctele specifice pentru detectarea semnalului ECG | * aprinderea ledului albastru * stingerea ledului roșu * începere transmisiune date pe interfața serial * afișare mesaj pe LCD | * Sistemul se comportă conform specificaților. |
| * Deconectarea electrozilor | * Aprinderea ledului roșu * Stingerea ledului albastru * Afișare mesaj pe LCD * Transmitere -1 pe interfața serială | * Sistemul se comportă conform specificațiilor |

Concluzii:

//TODO

Cod sursă:

// Add after real testing