# Poročilo o Vaji #2 – EKG

## Funkcija za iskanje T-vala

Pri EKG T val nastopi po S valu, zato ga iščemo med točko S in naslednjo točko Q. Funkcija je podobna funkciji za iskanje točke P z manjšimi spremembami.

Programska koda:

import numpy as np

#detekcija T valov

def detectWavePeaksT (iData, iIdxQ, iIdxS):

    oIdxT = iIdxQ[1:]

    for i in range(1, np.size(iIdxS)-1):

        od\_kje = iIdxS[i] #začetek pri S

        do\_kje = iIdxQ[i+1] # konec pri naslednjem Q

        for idx in range(od\_kje, do\_kje, 1):

            if iData[idx]>iData[oIdxT[i-1]]: #išči peak

                oIdxT[i-1]=idx

    oIdxT = oIdxT[:-1]

    return(oIdxT)

## Funkcija za določanje srčnega utripa:

Z uporabo vektorja časa lahko izračunamo razliko med časi udarcev za poljubne valove. Za izračun povprečne vrednosti in standardne deviacije uporabljamo funkciji np.average in np.std. Za izračun standardne deviacije frekvence je potrebno list z vrednostmi pretvoriti v array in ga predhodno deliti 1 z njim, saj drugače dobimo napačne vrednosti (ranga 50Hz).

import numpy as np

def computeHeartBeat(iTime, iIdx):

    Times = []

    oHBavg = 0

    oHBstd = 0

    oHFavg = 0

    oHFstd = 0

    for i in range (0, (np.size(iTime) -1)):

        cas = iTime[i+1] - iTime[i]

        Times.append(cas)

        i += 1

    oHBavg = np.round(np.average(Times), 3)

    oHBstd = np.round(np.std(Times), 3)

    oHFavg = np.round(1/np.average(Times), 3)

    oHFstdvred = 1/np.array(Times)

    oHFstd = np.round(np.std(oHFstdvred), 3)

    return oHBavg, oHBstd, oHFavg, oHFstd