

Technical report

Research utilities in Matlab for ANVERSO project: Continuous and Automatic analysis of Electrocardiographic Variables associated with the Risk of Sudden Death in patients with Brugada Syndrome. Elena Arbelo et al project. (by Pedro Gomis, version 2.0, January, 2023)

Data acquisition

ECG signals from Brugada syndrome (BS) patients were digitally obtained using a continuous 12-lead ECG recorder (Spiderview Plus, Livanova-Sorin group, ELA Medical, Montrouge, France) with high resolution mode, i.e., with 2.5 μV of voltage amplitude resolution (16 bits) and sampling frequency of 1000 Hz. ECG signals from the recorder flashcard were uploaded to Synescope analysis software (Livanova-Sorin group, Paris, France) which exports the ECG signals in ISHNE¹ format [1] for posterior analysis.

ISHNE format and digital signal resolution

The exported signals (name.ecg) from Synescope software were analyzed in MATLAB (Mathworks, Inc.; Natick, MA) after being converted from ISHNE format. The conversion was performed using, optionally, two freely available tools for MATLAB.

1. Firstly, the ecg-kit [2] was used to do the conversion². This toolbox returns an ECGwrapper MATLAB object containing the main information of the coded ISHNE header and a method to read the ECG signals. The header is presented following Physionet-MIT format style [3]. This software also computes detection of QRS and delineation of each ECG beat.
2. Secondly, a MATLAB m-file provided by Telemetric and Holter ECG Warehouse (THEW) was used³.

Briefly, ISHNE Holter format consists of a single binary file which contains a header followed by a data block having the ECG samples. The technical information contained in the header includes the lead specification, the lead quality, the amplitude resolution of each lead in nanoVolts (nV) and the sampling rate in Hz. The ECG data is stored using two bytes per sample per lead in samplewise order. For example, for 12-lead ECG recording each sampling data consists of 12*2 bytes block, following the lead order I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5 and V6. Each single lead sample of two bytes is stored in signed format, where the most significant bit is the sign and the analog-to digital (ADC) output code values range from -32768 to 32767.

The resolution of the Spiderview recorder is 16 bits. The resolution expressed in physical units (Volts) is 2.5 μV (2500 nV). Although ADC code values are dimensionless, the resolution can be

¹ ISHNE: International Society for Holter and Noninvasive Electrocardiology (<http://www.ishne.org/>)

² Available in: <http://marianux.github.io/ecg-kit/> ó <https://www.physionet.org/content/ecgkit/1.0/>

³ Available in: <http://thew-project.org/THEWFileFormat.htm> (see [read_ishne.m](#))

expressed in physical units per ADC code values or **ADC units**⁴. Thus, the resolution is 2.5 $\mu\text{V}/\text{ADC}$. This style is also used in [3] to describe the header information of [PhysioBank databases from Physionet](#) (known as MIT format) [4]. Since the manufacturer says that the resolution is 2.5 μV (2500 nV), it can be deduced that the amplitude range is = resolution * 2^{16} = +/- 81 mV.

ISHNE to MATLAB conversion

MATLAB code to read the ECG recordings is presented using either the ecg-kit software or read_ishne from THEW.

1) **Ecg-kit**. A block of 12 seconds of the 12-lead 24 hour ECG recording BH0005.ecg of the database was firstly processed to try the software. The first 30000 samples (30 sec) were left out since they correspond to a set of 1 mV reference pulse at each second. The 12 sec ECG was obtained from samples 30001 to 42000. The code used from the ecg-kit was (put proper path\name.ecg):

```
ECG = ECGwrapper('recording_name','G:\2018_ECG_BrugadaSyndrome\BH0005.ecg');
% Read the signal: ECG.read_signal(sample_ini, sample_end)
ECGs = ECG.read_signal(30001,42000); %
ECGs = double(ECGs); % signals in int16 --> float
gain = ECG.ECG_header.gain; % gain=4e-4 -> 1/2500 (Resolution = 2500 nV)
fs = ECG.ECG_header.freq; % Sampling frequency (2.5 uV)
baseline= ECG.ECG_header.adczero; % adczero = 0
[N,leads]=size(ECGs);
%Convert to Physical units (by default nV)
for i=1:leads
    ECGs(:,i) = (ECGs(:,i)-baseline(i))/gain(i);
end
% Units in nV (nanoVolts) --> mV (*1e-6)
ECGs = ECGs.*1e-6;
t=(1:N)/fs';
figure(1)
plot(t,ECGs(:,2)), grid, xlabel('t (s)') ylabel('mV')
title('BH0005.ecg - lead II')
```

⁴ ADC manufacturers also call ADC units as LSB (less significant bit) units.

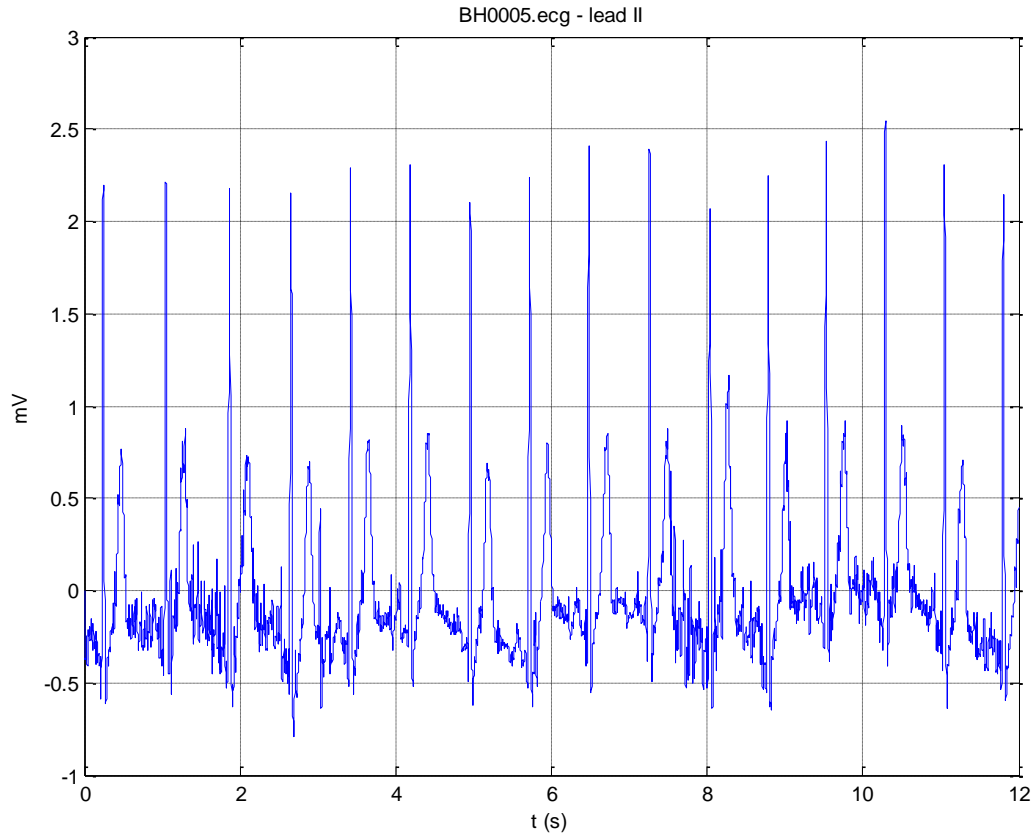


Figure 1. Lead II from a patient with record BH0005.ecg. Second 30 to 42.

The header is (green values added):

```
>> ECG.ECG_header
ans =
  nsamp: 86413248
  btime: '15:35:29'
  bdate: '1/1/2007'
  nsig: 12
  desc: [12x3 char] (I, II, III, aVR, aVL, aVF, V1..V6)
  gain: [12x1 double] (4.0e-04)
  freq: 1000
  adres: [12x1 double] (16)
  adczero: [12x1 double] (0)
  units: [12x2 char] (mV)
  rename: 'BH0005'
```

Figure 2 shows seconds 24-36: first 6 sec of lead III including previous 6 sec reference pulses.

```
ECGs = ECG.read_signal(24001,36000);
(idem)
ECGsIII = ECGs(:,3)-ECGs(450,3);
plot(t,ECGsIII), grid, xlabel('t (s)') ylabel('mV')
```

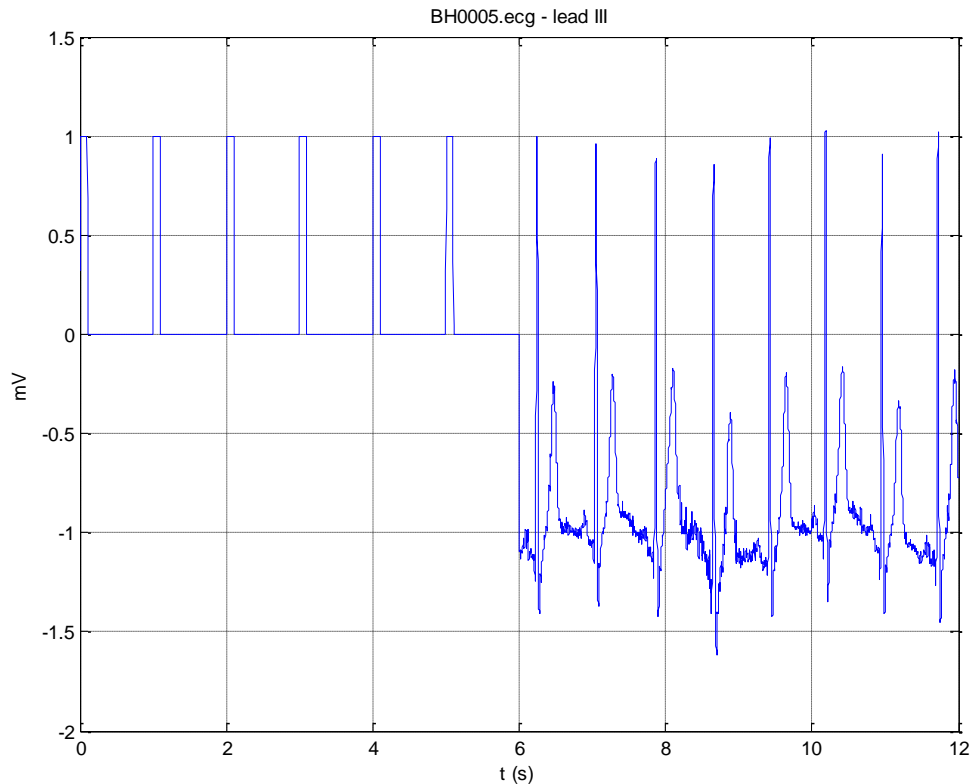


Figure 2. Lead III from patient with record BH0005.ecg. Second 24 to 36.

ECG signals were digitized at 16 bits with a resolution of $2.5 \mu\text{V}/\text{ADC}$. The ecg-kit toolbox created a header with MIT format, so the resolution was stored as ADC gain ($1/\text{resolution}$) and ADCres of 16 bits. ISHNE voltage amplitude units are given in nanoVolts (nV), thus,

$$\text{Resolution} = 2500 \text{ nV}/\text{ADC} \text{ (not saved in the header)}$$

$$\text{ADC gain} = 1/\text{Resolution} = 4.0\text{e-}4 \text{ ADC}/\text{nV} \text{ (saved in the header)}$$

The full scale input voltage range (Vfsr) is related to the resolution and the number of bits (16 bits). $\text{Resolution} = \text{Vfsr}/2^{16}$, thus

$$\text{Vfsr} = \text{Resolution} * 2^{16} = 2.5 * 2^{16} \mu\text{V} = 163840 \mu\text{V} = 163.84 \text{ mV} = \pm 81.92 \text{ mV}$$

2) **Function read_ishne.m from THEW.** The same 12 seconds block of the 12-lead 24 hour ECG recording BH0005.ecg was obtained from ISHNE file format using this routine in MATLAB from THEW. The code used was (put proper path\name.ecg):

```

% Function arguments: file name including path,
% initial time (seconds), Number of seconds
[hd,ecg] =
read_ishnel('G:\2018_ECG_BrugadaSyndrome\BH0005.ecg',30,12);
% Output in ADC units
Resol = hd.Resolution; % Resolution = 2500 nV each lead (2.5 uV)
fs = hd.Sampling_Rate; % Sampling frequency
% baseline= 0; % Not given, but it is the ISHNE standard
[N,leads]=size(ecg);
%Convert to Physical units (by default nV)
for i=1:leads
    %ecg(:,i) = (ecg(:,i)-baseline(i))*Resol(i);
    ecg(:,i) = ecg(:,i)*Resol(i);
end
% Units in nV (nanoVolts) --> mV (*1e-6)
ecg = ecg.*1e-6;
t=(1:N)/fs';
figure(2)
plot(t,ecg(:,2)), grid, xlabel('t (s)'), ylabel('mV')

```

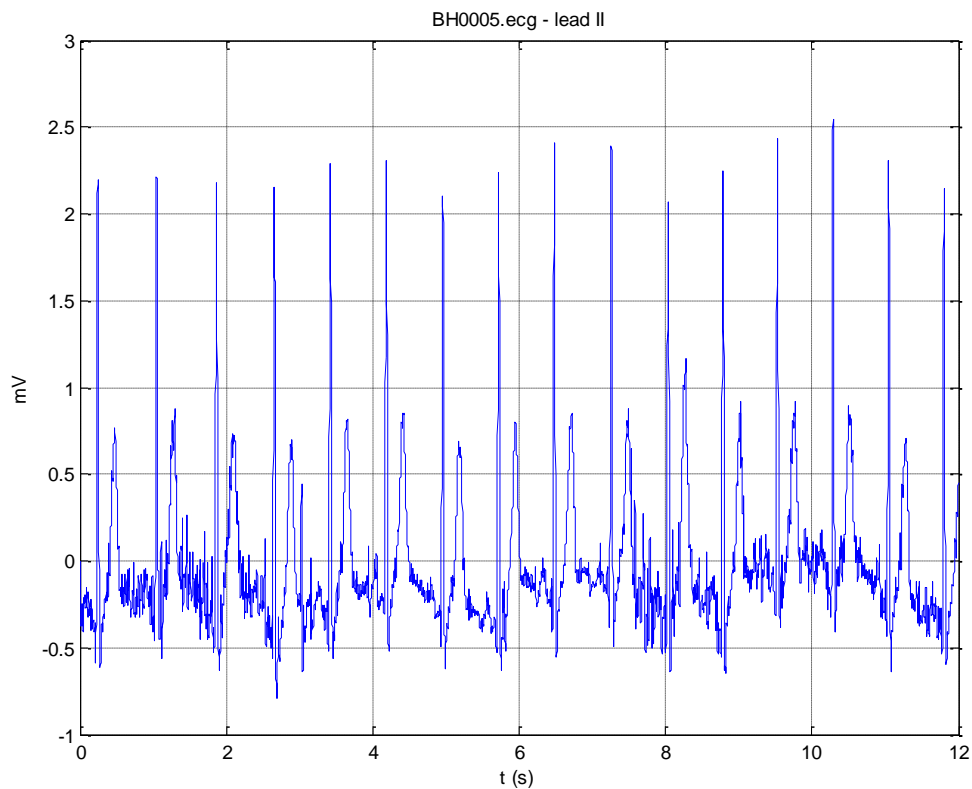


Figure 3. Lead II from patient with record BH0005.ecg. Second 30 to 42.

The header is (green value added):

```
>> hd
ans
    Sample_Size_ECG: 1.0370e+09
           nbLeads: 12
           Resolution: [12x1 double] (2500)
           Sampling_Rate: 1000

>> hd.Resolution(1)
ans =
    2500
```

Both options produce the same output ECG signals in MATLAB format and they were properly scaled in amplitude. Figure 4 shows a 6-second segment of leads I, II and III.

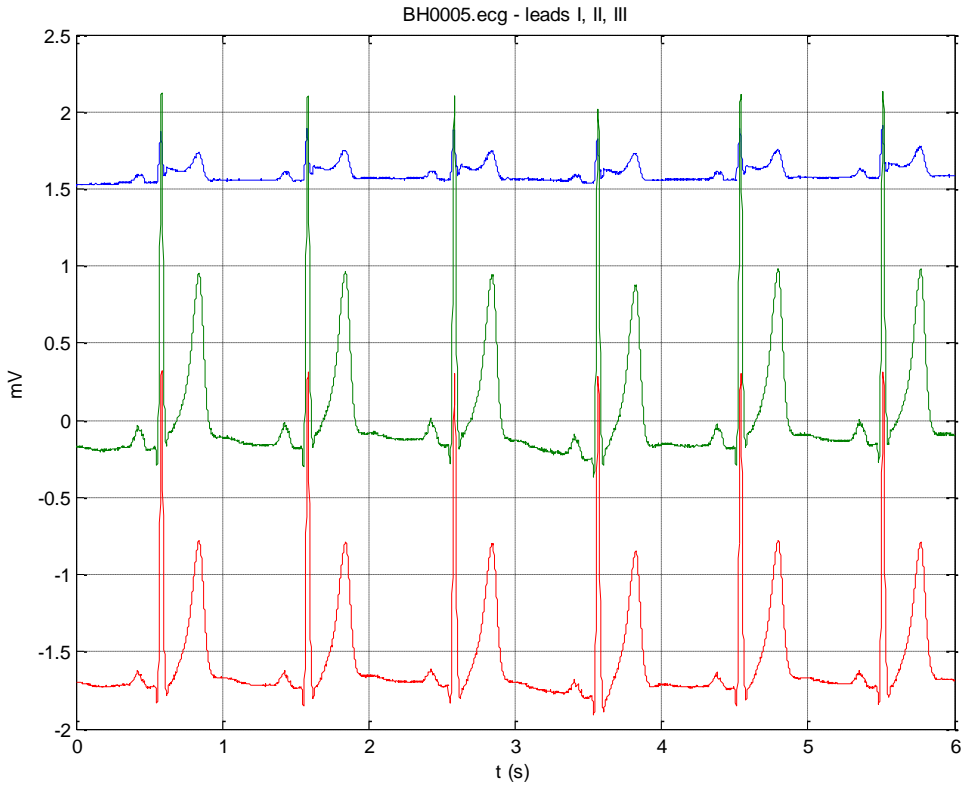


Figure 4. Leads I, II and III from patient with record BH0005.ecg.

A Synescope output print of the same record is shown in figure 5. Amplitude scale is shown in a 10 mm/mV square pulse.

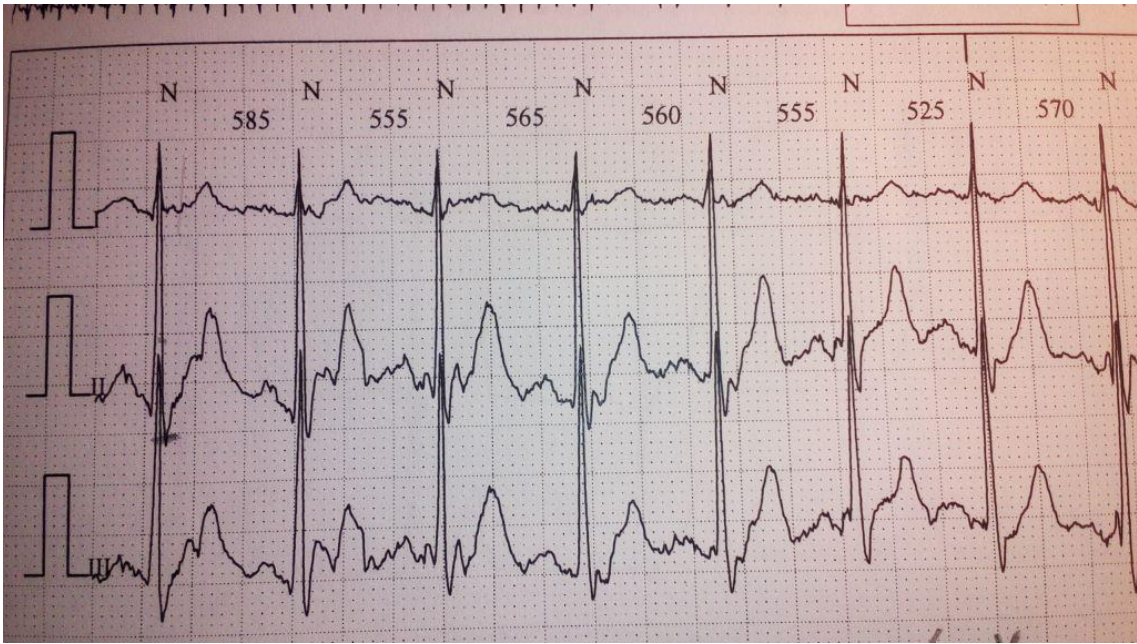


Figure 5. Snapshot from Synescope software. Leads I, II and III of record BH0005.ecg. Pulse of 1 mV amplitude (10 mm = 1 mV).

References

- [1] Badilini, Fabio and ISHNE Standard Output Format Task Force. The ISHNE Holter standard output file format. *Annals of Noninvasive Electrocardiology*, 1998, vol. 3, no 3, p. 263-266. http://www.amps-llc.com/uploads/2017-12-11/the_ishne_format.pdf
- [2] Demski AJ, Llamedo Soria M. ecg-kit a Matlab Toolbox for Cardiovascular Signal Processing. *Journal of Open Research Software*. 2016;4(1):e8. DOI: <http://doi.org/10.5334/jors.86>
- [3] Moody, George. *WFDB Applications Guide*. Tenth edition, Massachusetts Institute of Technology, Cambridge, MA, 2018. <https://physionet.org/physiotools/wag/>
- [4] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* 101(23):e215-e220 [Circulation Electronic Pages; <http://circ.ahajournals.org/content/101/23/e215>], 2000.