ECG LOGGER

Rhythmic Holter: “ECG Logger”

A Wearable Cardiac Monitor for Long-Term Data Acquisition and Analysis

Version 1.6

ENGLISH: The manual is only available in English.

FRANÇAIS : Le manuel n’est disponible qu’en Anglais.

中國語文： 用戶手冊僅提供英文版本。

Last update: May 11, 2018

ECG Logger is a Wearable Cardiac Monitor for Long-Term Data Acquisition and Analysis. The ECG Logger Project is aimed for providing a very low-cost open-source hardware and software for a Rhythmic Holter. The hardware has been made very simple and is based on an “Arduino Nano” with two companion boards for the SD card and the instrumentation differential amplifier. It helps monitoring arrhythmia syndromes such as bradycardia, tachycardia, extra-systoles or pause. In no case it can be use to replace a professional medical examination. Safety rules related to electrical devices must be complied and no connection shall be made between the Holter device and appliances connected to mains.

IMPORTANT: This project is provided for training or education and in no case for commercial purposes or medical diagnostics. It has not been approved for any kind of applications.

Notes, Remarks and Warning

Note related to use.

Important remark for avoiding failures or damages.

Critical note to avoid major injury or death.

IMPORTANT

Lethal conditions might occur when human bodies are in contact with high voltage. Therefore, under no circumstances a device shall be used while connected to an appliance powered by the main network.

Operating the device is under the entire, whole and sole responsibility of the user.

This project has been provided for educational purpose only and is not aimed for commercial applications neither it has been approved for any kind of uses.

Recycling

Please dispose electronics devices in an environmental friendly manner.

Copyrights

This project has been developed by AdNovea® in 2017. Respective contributions have been mentioned directly into the source code files. Please read the Licenses section for details.

TABLE OF CONTENTS

DESCRIPTION

ECG Logger is an affordable very low-cost (~35$) and simple ECG-Logger Holter offering a complete solution including a pocket-size ECG recorder hardware device with embedded firmware. The ECG signal is recorded onto a SD card memory at high frequency (250Hz sampling rate). The device is backed by the recent technology offering more freedom and accuracy for up to 24 hours continuous recording.

The system is based on an "Arduino Nano" microcontroller, an AD8232 Heart Rate Monitor amplifier board and a SPI SD card module with absolute minimum extra components

FEATURES

- Stable and reliable recorder, based on state-of-the-art and most recent technology
- Anti-jamming and anti-shock
- Small in size
- Waveform record and event marking
- Accurate start time record and sampled data
- Built-in SD card for storage (can be plug-in and pull-out)
- No need to compress original data, thanks to the large storage capability
- Faster USB 2.0 interface
- Record waveform details based on high precision and sampling frequency
- Record the state of pacemaker using higher sampling frequency.
- Automatic R-R peaks detection using Pan-Tompkins.
- International standard of 3 leads. Record up to 24-hour of ECG signal.
- Very simple and ergonomic user interface

IMPORTANT: This project is Open-Source licensed (*see licenses section*) for educational purposes only and none of its components can be used or reused for commercial purposes or applications.

USB driver must be installed on the computer prior to Holter (Arduino Nano) connection. Some Arduino use the FTDI chipset whereas some Chinese products require the CH340 chipset. The corresponding driver must be downloaded and installed on the computer. Click Start >> Control Panel >> Device Manager, and in the device list look for Com Ports (e.g. USB-SERIAL CH340).

SPECIFICATIONS (preliminary)

• Number of Channels	1
• Lead	standard 3-lead
• Sampling Rate	250 Hz
• Sampling Accuracy	10-bit / 8-bit selectable
• Recording Time	up to 24 Hours
• Time accuracy	+/- 1 minute per day
• Interface	USB 2.0 (230 kbauds)
• Scale Voltage	1 mV ±5%
• Sensitivity valve	≤20µV (TBC)
• Lowest voltage signal	50 µ Vpp (TBC)
• Input Impedance	≥ 1GΩ
• Input circuit bias current	≤ 0.1 µA
• Noise level	≤ 10 µ Vpp (0.1 Hz to 40 Hz)
• Common-Mode Rejection Ratio	>= 60 dB (DC to 60Hz)
• Electrode offset rejection	± 300 mV
• Voltage tolerance	± 500 mV
• Time constant	＞ 3.2 s (0.3Hz)
• Frequency response	0.05 ~ 150 Hz
• Filter	AC, EMG, Drift Filter, RFI
• Complex QRS detection	Pan & Tompkins algorithm
• Safety Human Body Model	8 kV ESD (HBM)

Physical Parameters:

• Type B*	internally powered
• Size (L x W x H) :	100 x 60 x 25 mm
• Net weight (w/o battery):	65 g
• Weight with batteries:	111g
• Total weight	195 g (incl. carrier/electrodes)
• Power	4 x AAA batteries
• Autonomy	>30 hours with Alkaline batteries

Average consumption:

• Normal mode	17 mA (Standby)
• Sleep mode	6.2 mA
• Recording mode	31 mA (~36 hours - depend on SD Card)

Package Content (*suggestion*)

- 1 x Holter recording device
- 1 x Patient cable (3-lead)
- 3 x Disposable ECG sensors
- 1 x Pouch with strap
- 1 x USB cable, 50 cm with ferrite
- 1 x Instruction Manual
- 1 x Downloadable application software

* *Type B is the least stringent classification, and is used for applied parts that are generally not conductive, where parts may be connected to earth, and can be immediately released from the patient.*

THE HOLTER DEVICE

The Holter device

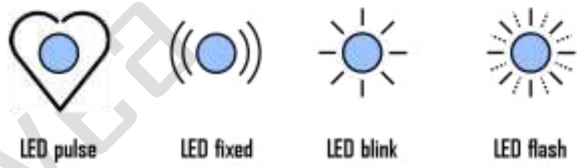
The electrode cable shall be plugged into the 3.5mm jack on the right side. The battery is switched on thanks to a key-switch on the right side. On the top are located the blue LED, the Event button and the USB socket.

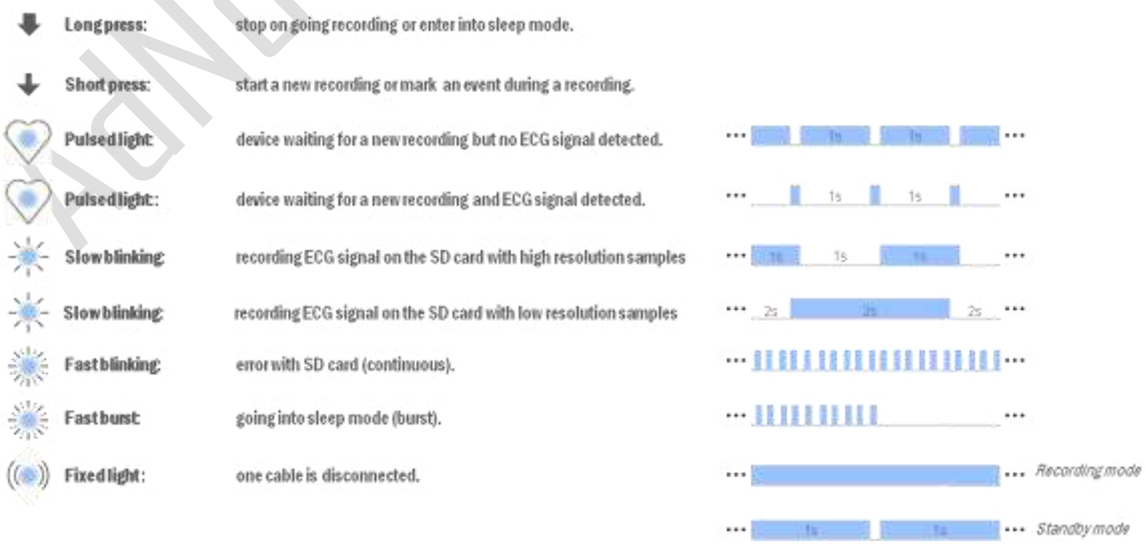
When connected to a PC with an USB cable, the Holter device is self-powered by the USB.

➲ Before the USB cable is unconnected; the Holter device must be switched on battery to keep memory of any changes in the configuration. Power-off erases any previous configuration.

LED indications

The Holter device status can be monitored by looking at the blue LED.





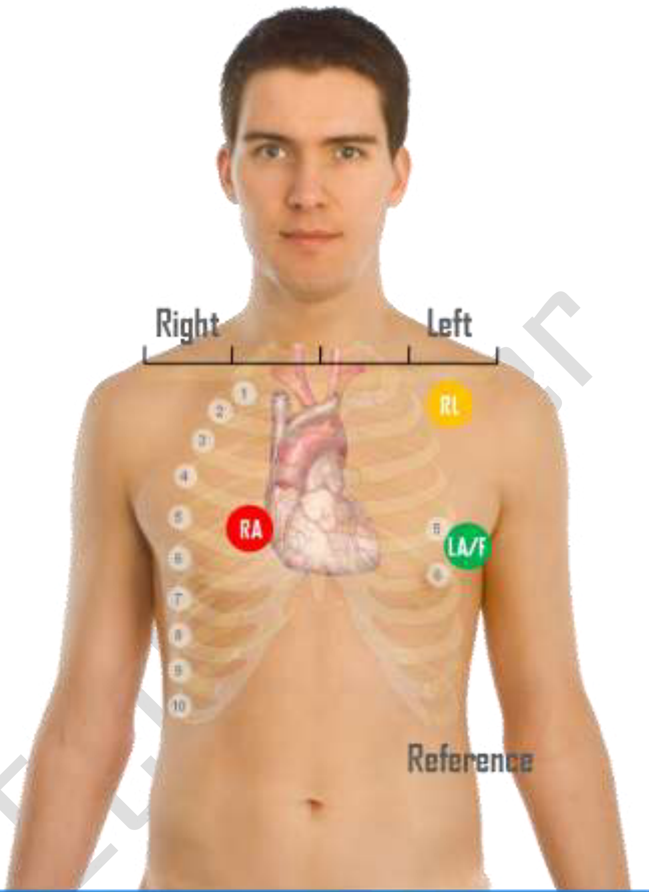
ELECTRODE'S PLACEMENT

For long duration Holter recording, electrodes on the chest are by far the best placement.

During the recording, the patient activities such as movements generate muscular noise and the mains power or electrical devices around are responsible for electronics noise making, the ECG signal impaired and unusable.

The electrode's placement and contact quality is crucial for ensuring the best signal to noise ratio. Remove hairs and clean skin areas where the electrodes will be placed. For better contact, add special gel or use electrodes with **solid gel and adhesive.**

ⓘ During nighttime, **wearing T-shirts** is strongly recommended for avoiding any undesirable interactions between the electrodes affecting the quality of the ECG signal (*parts of the body are causing short circuits by creation of multiple paths*). For example, contacts between left arm and thorax will divert the signal between electrodes LA/RL and contact between right arm and torso disturb the signal by creating parallel connections between the LA/RA or RL/RA.

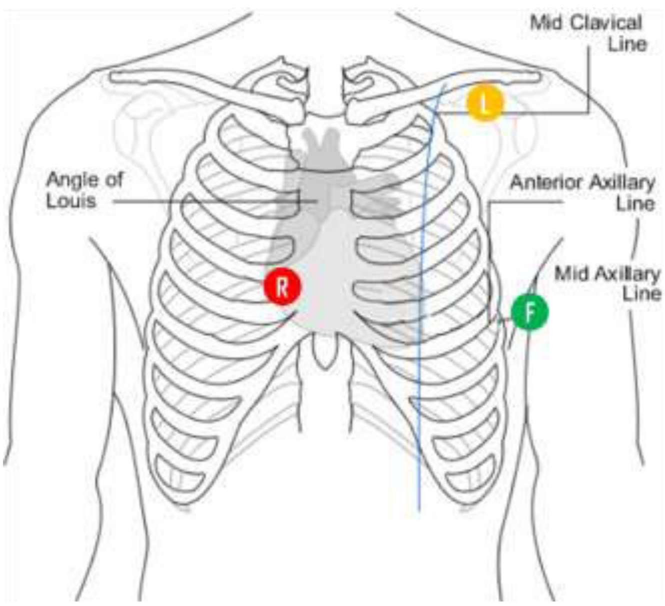


Electrodes	Electrodes Placement
LA (F)	Lowest rib on right side of chest, just above the hip
RA (R)	Middle to outside end of the right clavicle, close to the bone
RL (L)	In left infra-clavicular fossa, close to the bone

The **[F]** electrode takes place in the left infra-clavicle gap, close to the bone

From the angle of Louis, move your fingers to the right and you will feel a gap between the ribs. This gap is the 2nd Intercostals' space. From this position, run your fingers downward across the next rib, and the next one. The space you are in is the 4th intercostals' space. Where this space meets the sternum is the position for **[RA]** (V1).

Follow the line of the 5th intercostals space on the right side a little further until you are immediately below the centre point of the axilla, (mid-axilla). This is the position for **[RL]** (V6).



Electrode's inversions

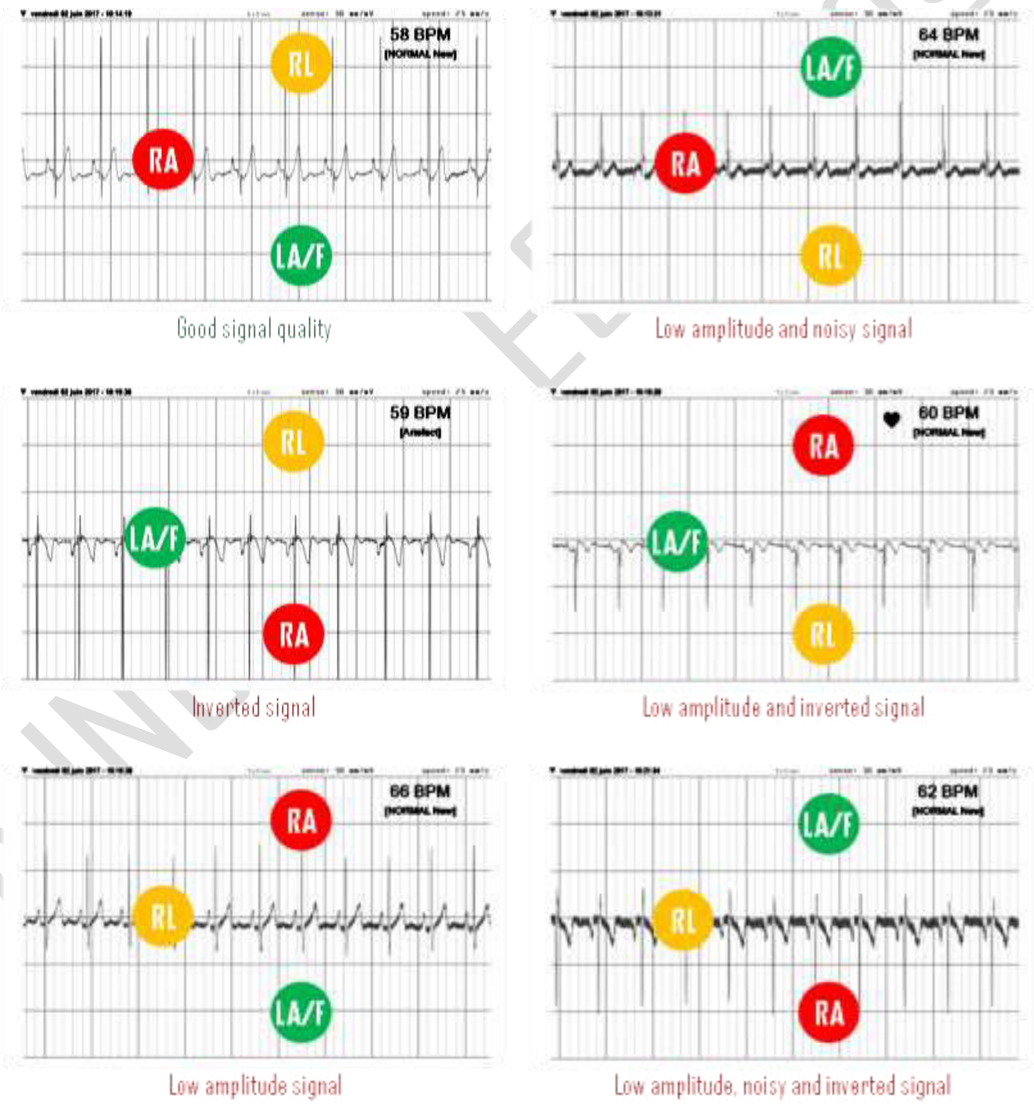
- F-RL reversal has almost no effect on the ECG shape but to increase the noise; you cannot identify it from the ECG and it has little effect on ECG interpretation. Nevertheless the SNR (signal to noise ratio) make Holter recording to be of lower quality.

- RA-F reversal inverts the ECG signal. The R-peak detection is impaired and classification cannot to be correctly performed.

- If the RL electrode has poor skin contact, ECG artifact may occur. A good clavicle contact is difficult to ensure due to the shape of the bone. This position is critical.

- F is used to minimize noise from common mode voltages, such as power lines parasitic

- Check RA-RL reversal for extremely low amplitude



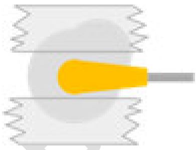
Impacts of electrode's inversion according to their relative positioning

LIVING 24 HOURS WITH A HOLTER

Monitoring cardiac activity shall not impact your day life and must ensure the Holter to be free from potential risks such as lead disconnection; electrode's tear off or cable breaks.

Lead connections

Once the electrodes are placed on the body, secure their position with two pieces of adhesive plaster.



ⓘ Avoid placing adhesive plaster directly over the cables or the plugs. It will be difficult to remove the adhesive residuals that will make the cable and plugs sticky forever.

Wearing the Holter

The Holter shall be inserted into the pouch and the strap placed around the neck. Preferably, wear clothes with collar in order to get rid of the pain of the strap directly in contact with the neck. The wires can go off the shirt either from the bottom or through the neck. The second possibility will be preferred as it prevents pulling too much on the cables.

The Holter is placed under the left shoulder and will remain quite stable despite the body movements. Enough freedom must be given to the cables to avoid traction on the cables or the electrodes.

During the day

It is not necessary to pay special attention to the Holter during the day or at night. Nevertheless, it's a good habit to mark specific events by pressing the Holter button. Events such as wake-up, breakfast, lunch, dinner, walk, sport activity, going to sleep or feeling pain or heart rate disturbances must be marked.

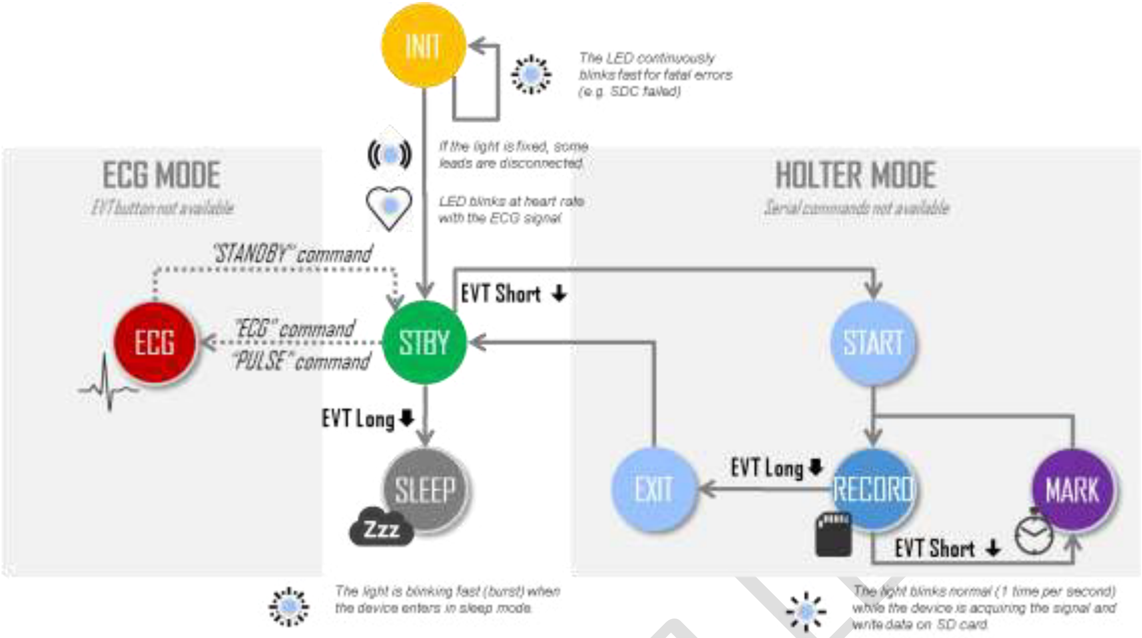
From time to time, monitor the LED that should be slowly blinking. The light toggles every second (in high resolution recording mode) or every other second (in low resolution mode).

During the night

At night, you can keep the Holter into pouch and have it on your chest rather than on the side. This is the preferred position if you sleep light or move quite a lot. If you sleep deep or do not move during the night, the preferred position is to place the Holter below the pillow and ensure the cables are free enough. This situation does not allow body rotations at night unless you are awake and can manage the cable when you move.

Bear in mind that the LED will be rather bright in the dark and could be annoying for light sleepers.

MODES OF OPERATION



ECG Mode

The device is connected to a PC (*powered on battery not by the mains*) to trace the ECG signal. The device can send either the ECG signal values (250 samples per second) or the beat rate (BPM beat per minute) on R-R peak detection (Pulse mode). The R-peak detection is done using the Pan and Tompkins algorithm. The algorithm also analyzes the ECG signal to identify normal, artifact (unstable signal) and arrhythmia.

HOLTER Mode

The device runs standalone and is only powered by the batteries. The recording time and duration has been previously configured using the ECG Logger Viewer application software. Place the device into a pouch and use a strap around the neck. The pouch can be placed under the shoulder for a better comfort during the daylight activities.

Remote Control

The ECG Logger Holter device can be controlled through a set of commands. These commands are sent and responses are readout from the USB-serial connection (*see the Device Firmware section for details)*.

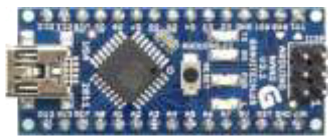
HOLTER DEVICE

The ECG Logger Holter device is made of 3 boards available from the selves, connected together in the simplest way as possible to avoid using specific tools or the need for additional components such as a custom printed circuit board (PCB). In the following section, the device assembly involves the 3x modules, a piece of Veroboard, 7x resistors, 3x capacitors, 1x LED, 2x switches and 1x battery coupler. About twenty five wires make the connection between all parts.

The **AD8232 instrumentation amplifier** used for the ECG signal acquisition also includes some analog processing for improving the signal quality. The amplifier provides a two stage amplifier with gains respectively equals to 100 and 11 for an overall amplification of 1100. The maximum differential signal amplitude is 2.7mVp-p before saturation. Then there is a two-pole high-pass filter with a 7Hz frequency cutoff for eliminating motion artifacts and the electrode half-cell potential. Additionally, the integrated op amp creates a two-pole low-pass filter using a Sallen-Key configuration for removing line noise and other interference signals with a cutoff frequency at roughly 25Hz. The circuit is also able to detect when one of the lead is off.

The **SD Card board** is used with a micro-SD card (2GB minimum) within the ECG Logger Holter device to provide the mass-storage unit for ECG data samples. The CATALEX board hosts a micro-SD card connector and is powered by +5V power supply. It uses the SPI protocol to communicate with the Arduino microcontroller. The SD Card must be FAT16 formatted. FAT32 formatted card are also supported but tend to run slower.

The **Arduino Nano** v3 is an ATMEL ATmega328 microcontroller. Its small form factor and the USB embedded port make it the preferred choice for the Holter device. It has enough digital and analog I/Os for the Holter application.

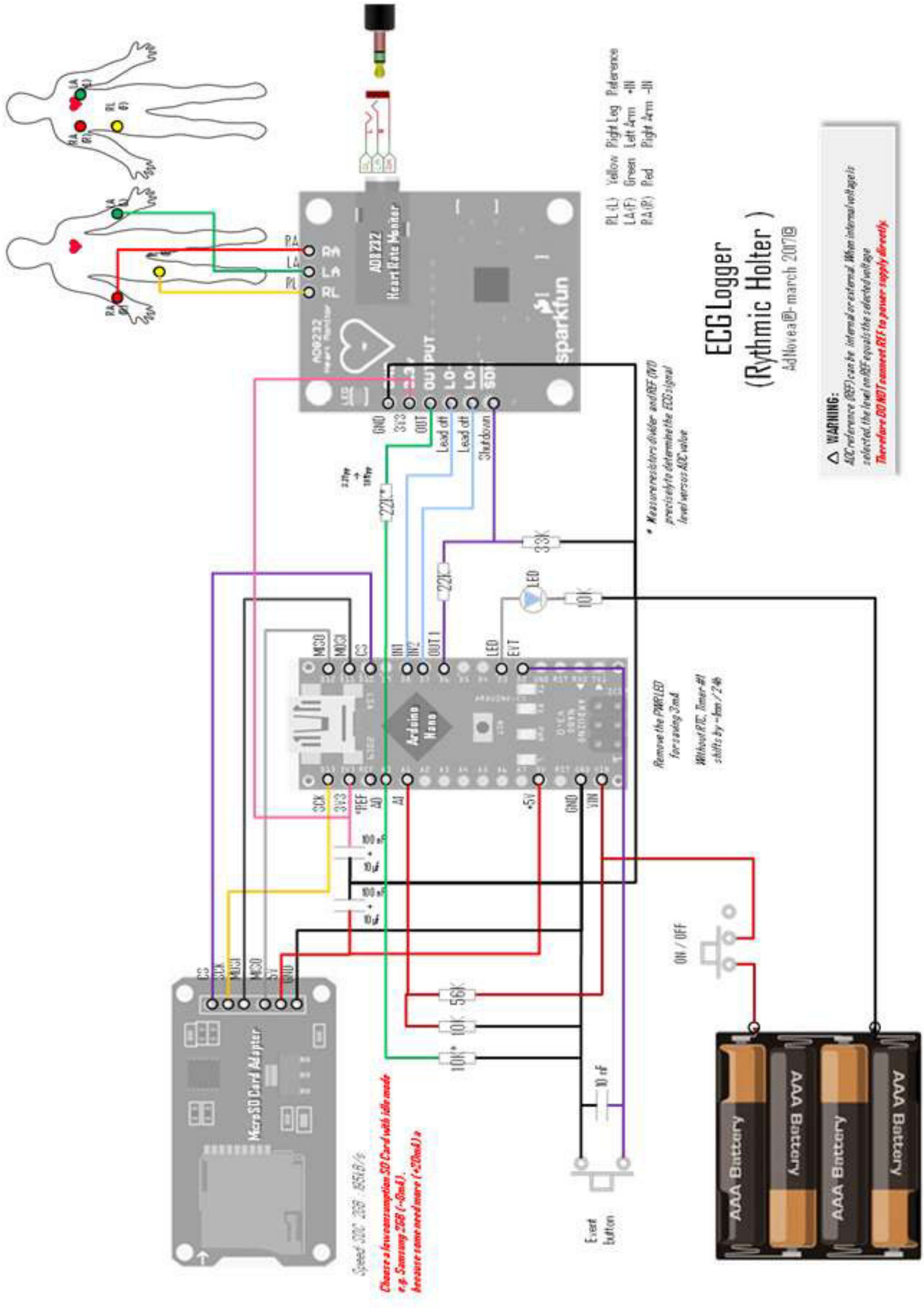


CHALLENGES

The challenges regarding the ECG Logger Holter device are:

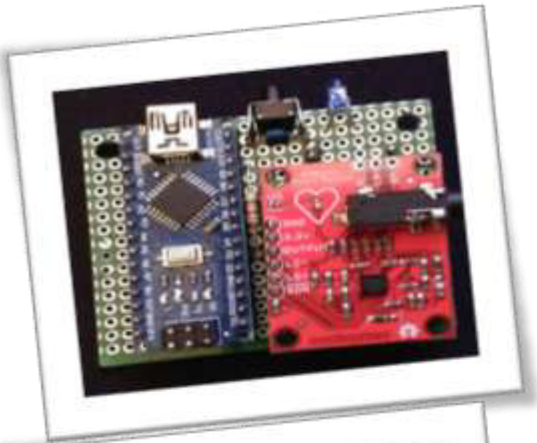
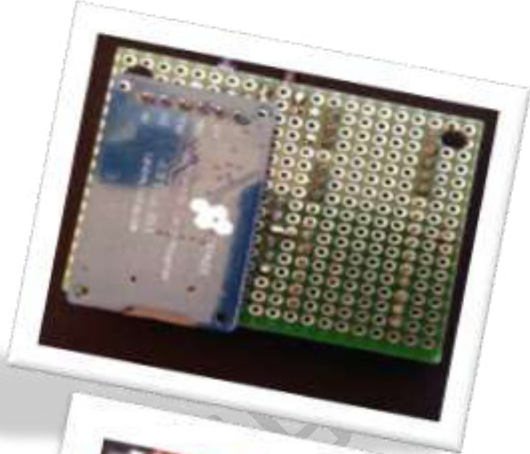
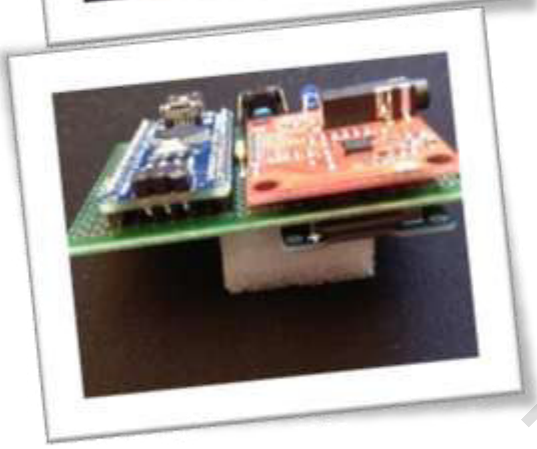
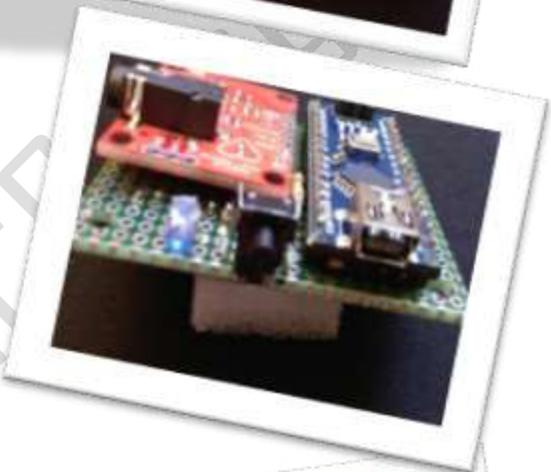
- **Form factor**: The Holter must be wearable during 24h; therefore, the size of the device shall remain as small as possible. The choice for a 100 x 60 x 25 mm plastic casing makes it standard regarding the size of commercial Holter. To make easier the assembly of the device, choice has been made to drastically limit the number of boards. Therefore no real time clock (RTC) will be available for time stamping which shall be made otherwise.
- **Consumption**: This is a major challenge because the Holter shall run on battery (rechargeable or not) during at least 24h. The choice of the SD card turned out to be an important element in the overall power consumption. Arduino power LED was removed for saving additional power.
- **Flash/RAM size**: The flash is limited to 30KB and the RAM to 2KB, which make the programming a challenge to fit the flash capacity and to keep RAM away from overflows.
- **Speed**: There are two challenges. The first was to make sure there will be no overruns while recording the ECG samples at 250Hz. The second was to take the best from the USB serial to download the data from the SDC as fast as possible. The best compromise between ECG file size and serial built-in speed has to be found. Unfortunately, we ended with 8-bit samples that creates 22MB/24h file size and takes 15mn for downloading.

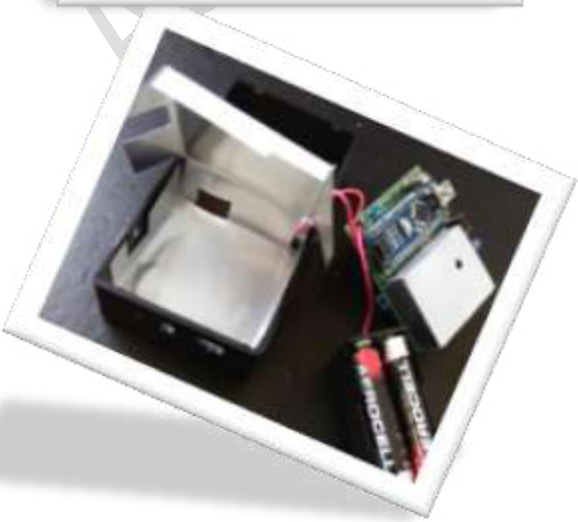
DEVICE SCHEMATIC



DEVICE ASSEMBLY

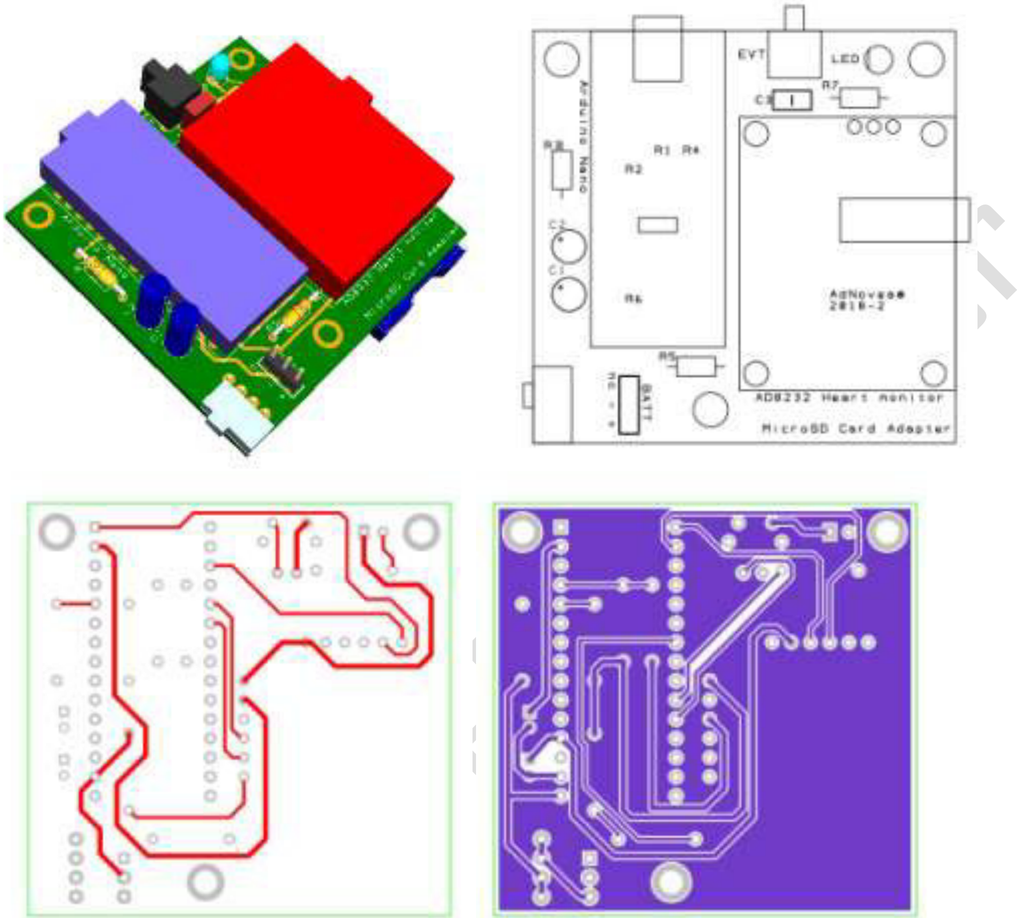
Check the electronic schematic above and the Bill of material section for components procurement and assembly. A veroboard PCB has been used for our “prototype”.

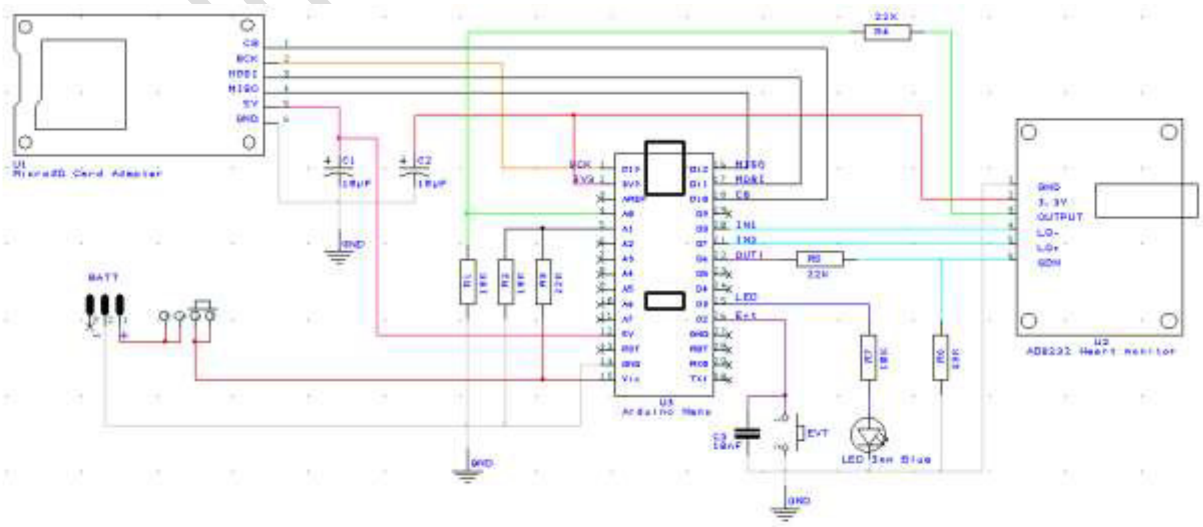




Printed Circuit Boards

Whereas veroboard and wires can be used for assembling the Holter device, it may also be done using a double layer PCB.

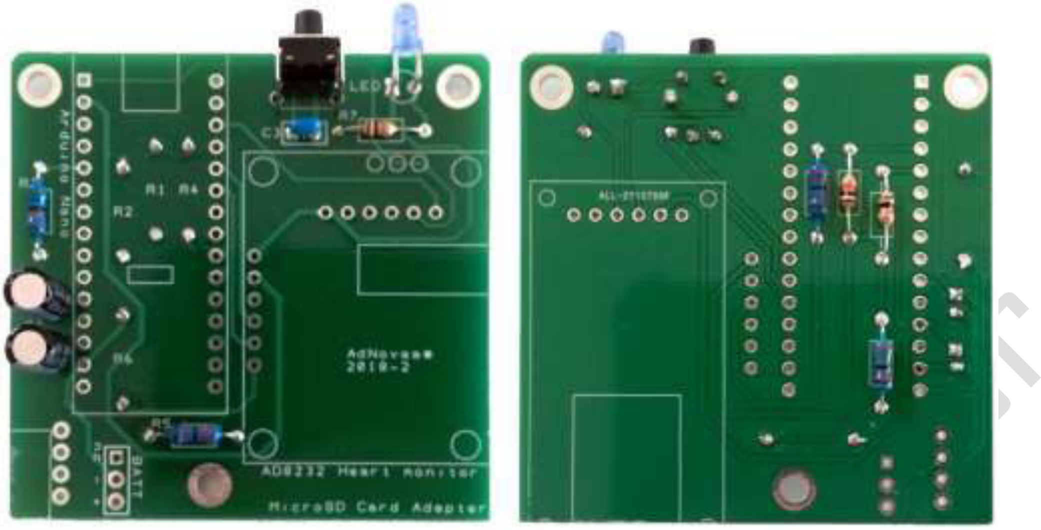


PCB: 55.88 x 54.61 mm / FR4 / thickness 1.6mm - Top & Bottom copper layers



« ECG Logger » holter device schematic

PCB Assembly Steps



The first step is too sold the seven resistors, three capacitors, the LED and the tack button.

➲ The next step is to install the **SD Card reader** and then the **AD8232 heart monitor** boards otherwise it won't be possible to solder the SD Card reader. Finally you can weld the **Arduino Nano** board, the power switch and the battery holder.

The battery cable can be secured using a nylon cable tie.

ECG Logger assembled

List of components

Ref.	Qty	Component	RS P/N	Description
U3	1	Arduino Nano		Atmega328
C1	1	Capacitor 10uF/50v	228-6874	KS radial Al electrolytic cap
C2	1	Capacitor 10uF/50v	228-6874	KS radial Al electrolytic cap
C3	1	CAPPTH2		Capacitor Standard
PL1	1	CONN_SIL_3		3 way Pin Header
U2	1	Heart monitor		AD8232 Heart monitor
LED	1	LED 3mm Blue		Blue LED
R1	1	Resistor 10K		Resistor 1/4 W
R2	1	Resistor 10K		Resistor 1/4 W
R3	1	Resistor 22K		Resistor 1/4 W
R4	1	Resistor 22K		Resistor 1/4 W
R5	1	Resistor 22K		Resistor 1/4 W
R6	1	Resistor 33K		Resistor 1/4 W
R7	1	Resistor 10K		Resistor 1/4 W
U1	1	SD Catalex		MicroSD Card Adapter
SW2	1	Slide jumper	204-7871	Slide switch
SW1	1	Tact 90		Micro switch 90°

ⓘ *PCB Gerber and drill files are available from the ECG Logger Download section at SourceForge.*

Device enclosure cuttings

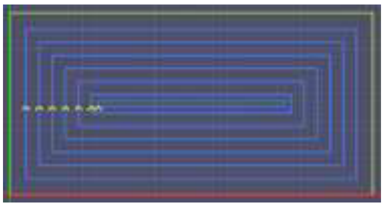
For milling the enclosure, we have used a CNC machine driving through G-Code scripts.

The selected 2x2 AAA-battery holder is slightly too high for the selected plastic enclosure. Two pockets (52x30 mm) need to be mill on the top cover and the bottom of enclosure. The depth will be 1 mm for the enclosure bottom and 0.5 mm for the top cover.

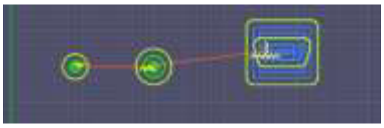
- *An alternative consists of opening a hole on the top cover and then sticking the label.*

The USB socket, Event's button and blue LED must fit into the enclosure. The distances are respectively:

- LED – Button = 11 mm (centered)
- Button – USB = 148 mm



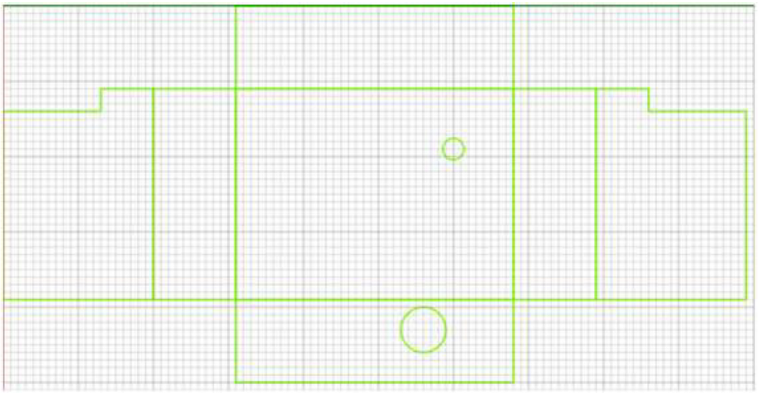
The Led and button holes are aligned with bottom of the USB socket. An extra pocket could be required around the USB socket if the cable connection is not well secured.



Depending of the power switch you will have used, the hole for the one suggested is 7 x 2.5 mm.

AD8232 Heart Monitor shielding

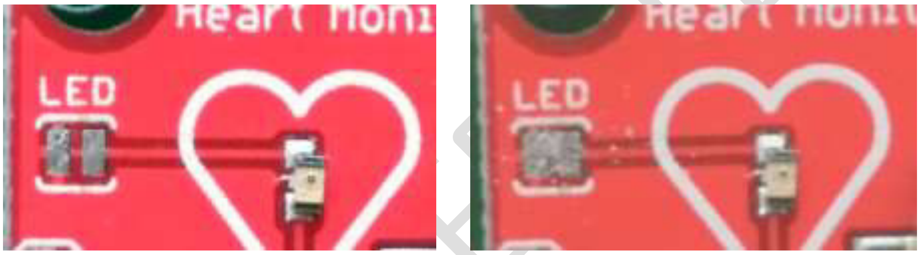
It does not seem mandatory to shield the AD8232 board. But if you want to make a shielding, a form can be folded as shown below using a sheet made of cookware aluminum laminated into a plastic pouch. Extremities below are fixed with adhesive tape.



ED8232 board shield (100 x 50 mm)

Power consumption optimization

Disabling the AD8232 Heart Monitor LED



The LED pad

The pad is short-circuited. Therefore, the connection between the two pads shall be cut with a cutter. Should you need to re-enable the LED, just put solder on the two pads to reconnect the LED.

Disabling the Arduino Nano POW LED

The Arduino Nano power LED is directly connected to the power supply and light on as soon as the microcontroller is powered. The LED consumption will reduced the battery light and is not required when the Holter is running. To remove the **POW LED**, you must use a soldering iron. Be cautious when removing the LED to avoid overheating or to create short-circuits due to solder remains. This will save about 3.90 mA.

Choose the right SD Card

Some SD card need more current than others. See the “Battery life” section.

DEVICE FIRMWARE

The Arduino Nano is programmed using the open-source provided "ECG Logger" sketch. The SdFat and Time-**master** libraries are required with the ECG logger software. For better legibility, the source code has been split in 4 files:

- ECG_Logger_v15.ino — Main routine. Source to be loaded
- ECG_Logger.h — Header file with all constants and variables
- ECG1_ctrl.ino — Routines for HMI and device control
- ECG2_qrs.ino — Routines for the ECG peak detection and classification
- ECG3_sdc.ino — Routines for managing the SDC recording

The ECG Logger device can be controlled using a console (*Arduino monitor, Termite terminal or ECG logger viewer console*) with the set of commands below. The serial communication baud rate is 250,000.

ⓘ Despite the serial baud rate of the Arduino Nano can be theoretically set up to 2Mb/s, the actual data speed transfer will never exceed 230 kb/s and can be achieved by selecting the 250kb/s baud rate.

Modes:

- **ECG** — Mode ECG Live (outputs ECG HiRes values on serial Comms)
- **PULSE** — Mode PULSE (outputs R-R peaks with beat rate, type and distance)
- **STANDBY** — Return to standby mode

Data:

- **CLEAR** — Prepare for new recording (*erase all previous files*)
- **T**<epoch> — Set UTC Epoch time (*ex. T1489828532*)
- **HOUR** — Display date/time (*YYYY-MM-DD HH:MM:SS*)
- **X**<secs> — Recording duration in seconds (*86400 for 24h by default*)
- **HIRES** — 10-bit data (Holter mode with R-R peak detection)
- **LORES** — 8-bit data (Holter eco mode - less SD space)
- **FILTER**<on/off> — Enable/disable ECG noire reduction filtering

Files:

- **LIST** — List files on the SDC (*date and size*)
- **R**<no> — Delete files (<no>= __, 00 ... 99)
- **D**<no> — Dump file (<no>=__, 00 ... 99). Binary is preceded with file size
- **NBRECS** — Return the number of recordings (*next available number*)
- **FREE** — Free space on SD Card (*in GB*)

Miscs:

- **INIT** — Reset of the firmware (*soft reset*)
- **?** — See manual reminder
- **K**xxxx — Set serial port baud rate (xxx: 1,200 ... 2,000,000)
- **BATTERY** — Read battery level (*in mV*)
- **VERSION** — Return current firmware version
- **PARAMS** — Set device algorithm parameters (*; separator*)
  - Batt_Convert — in mV – default is 6944
  - Max_HF — [0-1023] – default is 10
  - Min_DC — [0-1023] – default is 0
  - Max_DC — [0-1023] – default is 400
  - Min_ADC — [0-1023] – default is 150

- Max_ADC [0-1023] – default is 1023
- Refractory [0-999] – default is 68 samples
- Cond_BRADY [0-999] – default is 1091 ms (55 BPM)
- Cond_TACHY [0-999] – default is 600 ms (100 BPM)
- Cond_EXSYS [0-99] – default is 70%
- Cond_PAUSE [0-999] – default is 2000 ms

PARAMS;6944;10;0;550;300;1023;68;1091;600;70;2000

Default parameters

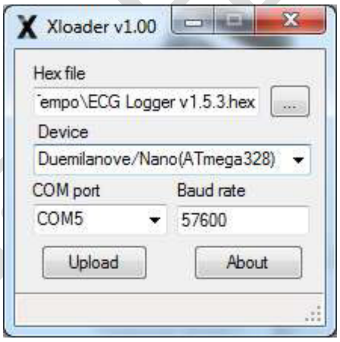
Without programming, the Holter mode will record the ECG signal during 24h with 8-bit samples. The date will be set to January 1st, 1970.

DEVICE PROGRAMMING

Device programming can be done using either with:

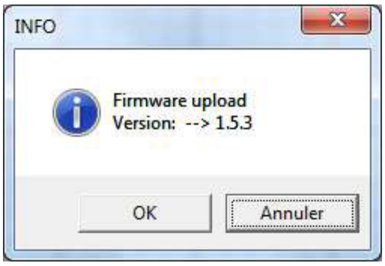
- the Arduino ECG Logger device source code and the Arduino IDE

- the ECG Logger device HEX file and the Xloader utility. "Xloader" application can be downloaded from the author website at http://xloader.russemotto.com/
  Select the "*Duemilanove/Nano (ATmega328)*" device, the serial port where the device is connected to and browse to the HEX file. Click on "Upload" to proceed with the device programming.

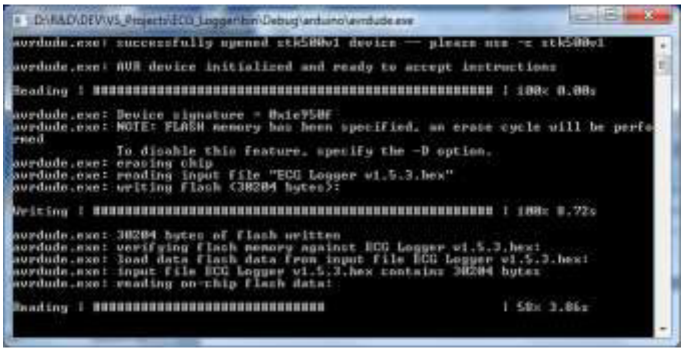


- Device programming is also possible from the ECG Logger Viewer application software.
  Start the ECG Logger Viewer application and select from the menu "*Device > Firmware upload*". Assuming the serial com port in the preferences matches the port used by your device, just depress the OK button to start the firmware upload. The new firmware version is shown in the message box.

  ⓘ Check/set correct device port in "*Tools >> Preferences / Device > Com_port*" (e.g. COM5).

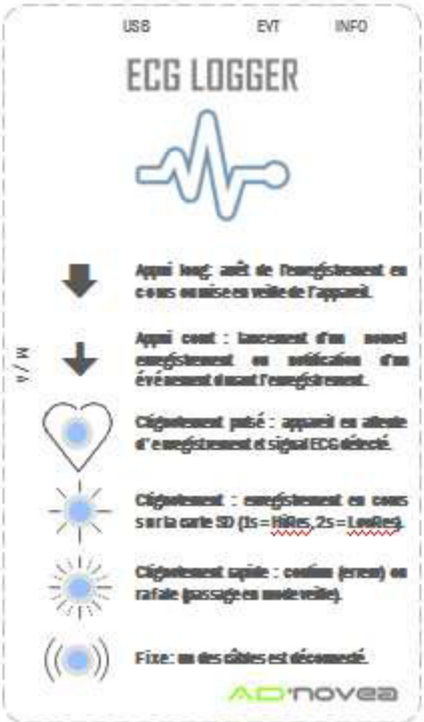


DO NOT CLOSE the progression window otherwise the device firmware will be corrupted.

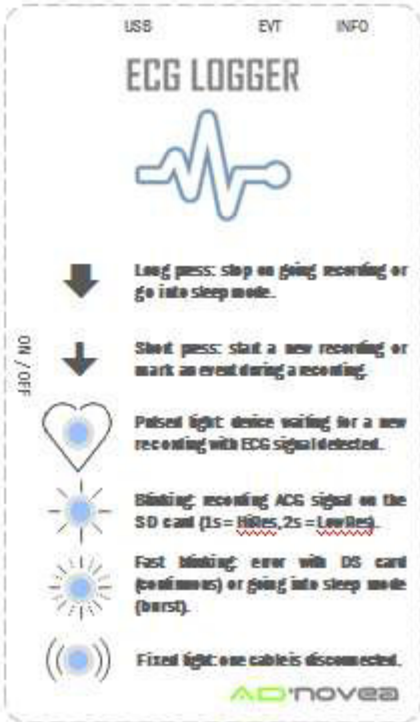

D:\R&D\DEV\VS_Projects\ECG_Logger\bin\Debug\arduino\avrdude.exe
avrdude.exe: successfully opened stk500v1 device -- please use -c stk500v1
avrdude.exe: AVR device initialized and ready to accept instructions
Reading | ## | 100% 0.00s
avrdude.exe: Device signature = 0x1e950f
avrdude.exe: NOTE: FLASH memory has been specified, an erase cycle will be perfo
rmed
To disable this feature, specify the -D option.
avrdude.exe: erasing chip
avrdude.exe: reading input file "ECG Logger v1.5.3.hex"
avrdude.exe: writing flash (30204 bytes):
Writing | ## | 100% 8.72s
avrdude.exe: 30204 bytes of flash written
avrdude.exe: verifying flash memory against ECG Logger v1.5.3.hex:
avrdude.exe: load data flash data from input file ECG Logger v1.5.3.hex:
avrdude.exe: input file ECG Logger v1.5.3.hex contains 30204 bytes
avrdude.exe: reading on-chip flash data:
Reading | ############################# | 58% 3.86s

DEVICE STICKER

For the highest level of durability, stickers are made of color printed sheets laminated into a plastic pouch and stick on the enclosure cover using double-sided adhesive tape.



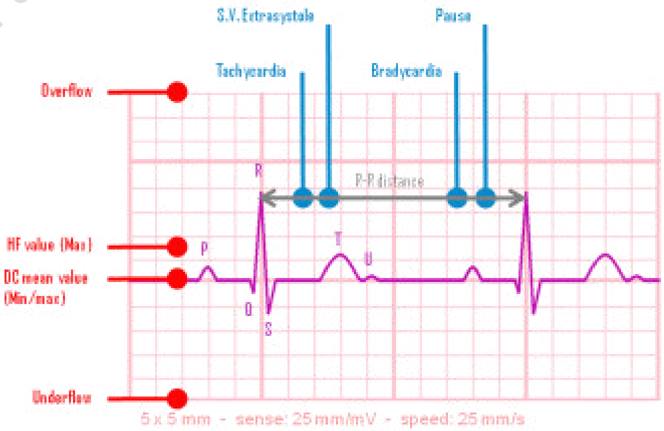
Français



English

ECG SIGNAL INTERPRETATION

The naming conventions P-Q-R-S-T-U for the ECG signal specificities are given on the diagram below. The grid, paper speed and sense amplitude may be different from the figure below and is used for measuring the time on the X-axis direction and the amplitude on the Y-axis direction.



The ECG signal is made of the QRS complex and additional side events such as:

- The **P** wave represents atrial depolarization.

- The **QRS** complex represents ventricular depolarization.
- The **T** wave represents ventricular re-polarization.
- The **U** wave represents papillary muscle re-polarization.

Pulse rate is given in Beat per Minutes using the distance between two consecutive R-peaks.

The firmware and application software will try to get rid of unwanted parasitic such as muscular noise and electrical interferences. Moreover, the position of the R peak will be detected and corresponding samples marked accordingly. Finally, some additional processing can be performed on the signal to classify the ECG signal using the sample values. Thresholds on the HF and DC signals as well as overflow and underflow detection will enable to detect out of range signal and classify the corresponding samples as Artifacts.

Complementary analysis will also be performed on the values of the R-R distance to determine out of range peaks but also conditions for arrhythmia disturbances such as extra systoles (shorter distance), pause (longer distance, tachycardia (fast rate) or bradycardia (slow rate).

More information can be found at https://en.wikipedia.org/wiki/Electrocardiography

Signal output level

The signal from the electrodes is applied on the inputs of the instrumentation AD8232 board for differential amplification.

- The overall gain = 1100 given from the in-amp gain=100 and the op-amp gain=11
- The maximum AD8232 input voltage = 2.7mV and the corresponding maximum output voltage = 2.97V

In the ECG Logger Holter, the AD8232 board ECG output signal is divided by x0.306

- Ratio between the resistors of the divider bridge (22.23K/9.80K)
- Maximum ECG signal at the divider output = 1.007V

NOTE: the resistor values precision determine the divider ratio that must be correctly setup into the application software parameters.

The Arduino Nano internal ADC convert voltage reference is set to: 1V1

- Reference voltage precision is VRef +/-3%
- Actual measured value is 1.068V for the 1V1 internal reference

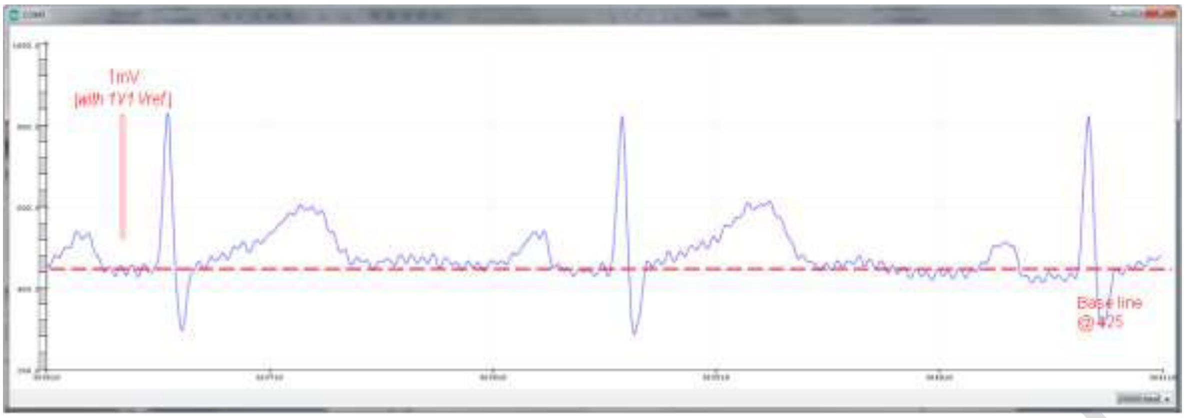
The High Resolution (HiRes) mode enables the Arduino Nano ADC to provide 10-bit samples

- Number of levels = 1023
- Resolution of the ECG signal = 1.044mV
- 1mV ECG input signal → 373mV ADC input signal → 1mV = **357 levels**

In HiRes mode, there are 250 samples per 512B block and every block contents 1 second of recording. One day of recording equals: 24h = 86,400 seconds and gives à 43MB file

- 24h → 10-bit → 86400 blocks of 512B→ 1 file → 44,236,800B → 43MB

The Low Resolution (LoRes) mode provides 8-bit samples (256 levels). There are 500 samples per 512B block. One day recording gives à 22MB file

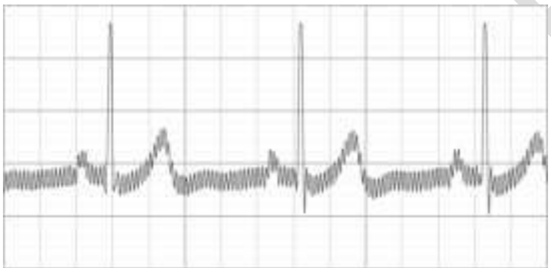


One can notice some residual 50Hz noise superimposed over the ECG signal

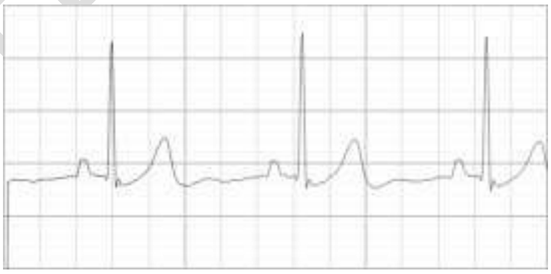
Impacts of the noise

When the Holter device is connected to the computer, the electronics noise is rather important and the mains interference (*50Hz on the figure below*) is significant. Remember that the signal to noise ratio is greatly impacted by the quality of the electrode's contacts and their correct positioning. The Pan & Tompkins band pass filter can strongly improves the quality of the ECG signal and get rid of the 50/60 Hz noise as well as the muscular noir and base wandering (*mean level*).

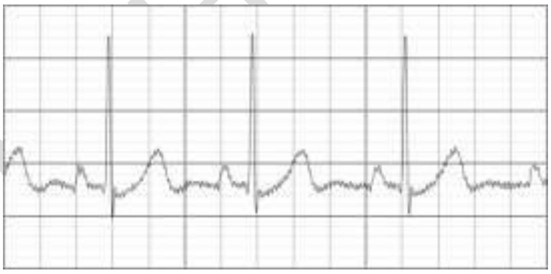
When the Holter runs on the battery, the remaining mains noise comes only from the electromagnetic component of the radiating field which is captured.



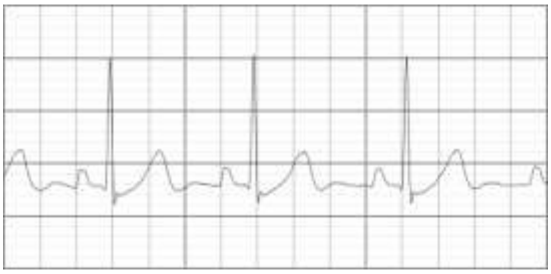
Device connected to computer



Signal band-pass filtered



Device disconnected from computer



Signal band pass filtered

PROCESSING FOR QRS AND ARRHYTHMIA DETECTION

The Arduino firmware includes some signal processing for noise reduction and to detect the R-peak and classify the rate disturbances.

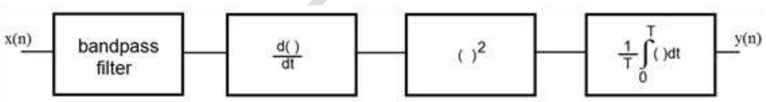
ⓘ Algorithms use data buffer that introduce delay. Therefore the R-peak detection delivers information about 25 seconds after the start of the ECG signal acquisition.

NOISE REDUCTION:

For improving the quality of the ECG signal before R-peak detection and recording, a digital noise reduction filter has been implemented. It can be switch off using the FILTER OFF/ON command from the Console. This filter drastically improves the quality of the signal by filtering High Frequency interferences. By default the noise reduction filter is enabled at startup.

PAN and TOMPKINS algorithm filters and methods:

- DC mean compensation (optional)
- Low pass filtering
- High pass filtering
- Derivative
- Squaring
- Integration
- Adaptive thresholding



Pan & Tompking algorythm

Classification of R-R peaks uses:

- For Artifacts:
- Min/Max overflow
- DC level range
- HF max amplitude
- For Arrhythmia
- BPM = 70% previous → ESSV
- BPM = +15% → Pause
- BPM > 110 → TACHY
- BPM > 50 → BRA

Classification of peaks (Arrhythmia):

- ARTIFACT 0
- NORMAL 1
- ESSV 2
- ESV 3 (not used)
- PAUSE 4
- NORMAL_N 5 (first Normal peak after artifact or start)

- BRADY 6
- TACHY 7
- ESS 8 (not used)
- FA 9 (not used)

Peak s detection:

In High resolution mode (HiRes), the 10-bit data samples corresponding to an R-peak are marked. Their bit-15 is set to 1 whereas the 10 least significant bits (LSB) are used for storing the signal value. The R-R distance between two successive peaks and the peak's type resulting from the peak classification are recorded into each bloc header containing 1 second of sampling.

In Low Resolution mode (LoRes), the 8-bit samples corresponding to the R-peaks cannot be marked because the 8 bits are used for storing the signal value. The R-R distance between two successive peaks and the peak's type resulting from the peak classification are recorded into each bloc header containing 2 second of sampling. Because there will be more than one peak per slot of 2 second, only the R-R distance and type values corresponding to the latest peak occurring in the 2 second time slot can be stored within each block.

Artifacts detection:

We measure the amplitude of the band pass filtered ECG signal, the mean DC level as well as the overflows and underflows to determine if the ECG signal is consistent or not. Irrelevant signal is marked as artifacts. The R-peak detection may require a couple of QRS complex before the stabilization.

BATTERY LIFE

Battery AAA

- R03: Carbone/Zinc – 540mA/h (***insufficient*** *capacity*)
- LR03: Alkaline 1.5V – 860 - 1200mA/h
  Below 4.2V alkaline batteries are weak
- HR03 NiMH 1.25V - 600 - 1250mA/h

Measured current (*PWR LED removed*):

- Standby 16.90 mA
- Sleep 9.13 mA
- Recording 25.80 mA

Measured overall current along with SD card types (*recording*):

- Mobizen Micro SD 4GB 46 mA
- Samsung 2GB 27 mA (but poor quality with 20 to 25 overruns per minutes)
- SanDisk Ultra 8GB 31 mA (recording : 48.35 mA)
- SanDisk 4GB class 4 31 mA
- SD card stop working around 3.8-3.9V
- ECG logger min. voltage has been set to 4V to go into sleep mode while recording

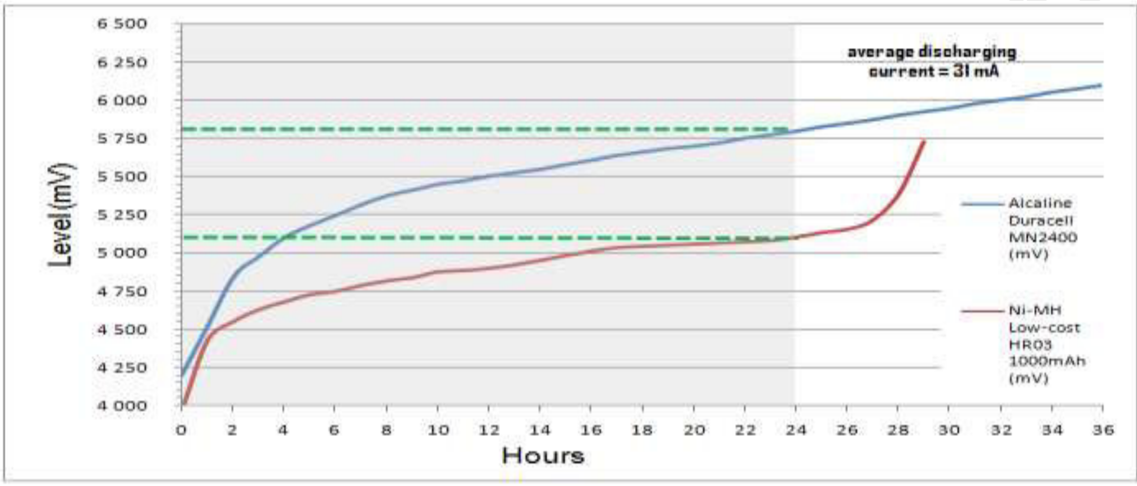
Battery levels:

- **LR03** Full capacity 6 050 mV
- **LR03** Empty level 4 125 mV (1.05V by battery element)
- **HR03** * Full capacity 5 950 mV
- **HR03** *Empty level ~4 150 mV
- **USB power** 4 210 mV

* 1000mAh / 1.2V Ni-MH

Battery level measurement accuracy

The selected ADC internal reference is set to 1V1 (actually1.068V) and provide a 1023 digital value for 1.1V ADC signal. Battery level measurement is performed trough a divider bridge made of 56K and 10K resistors (ratio is 1/6.6) and applied on the A1 ADC input. Four fully charged battery is about 6V DC level and delivers 0.909V through the divider and give a digital value about 845. In “ECG_Logger.h” file, the “BAT_CONVERT” variable converts the digital value into mV reading and should be set to 7.100. Due to resistor (10%) and reference voltage tolerances the conversion can be slightly different. We have set “BAT_CONVERT” to 6.944 but it can be adjusted for your specific design. Change of the variable requires recompiling and uploading the firmware again.

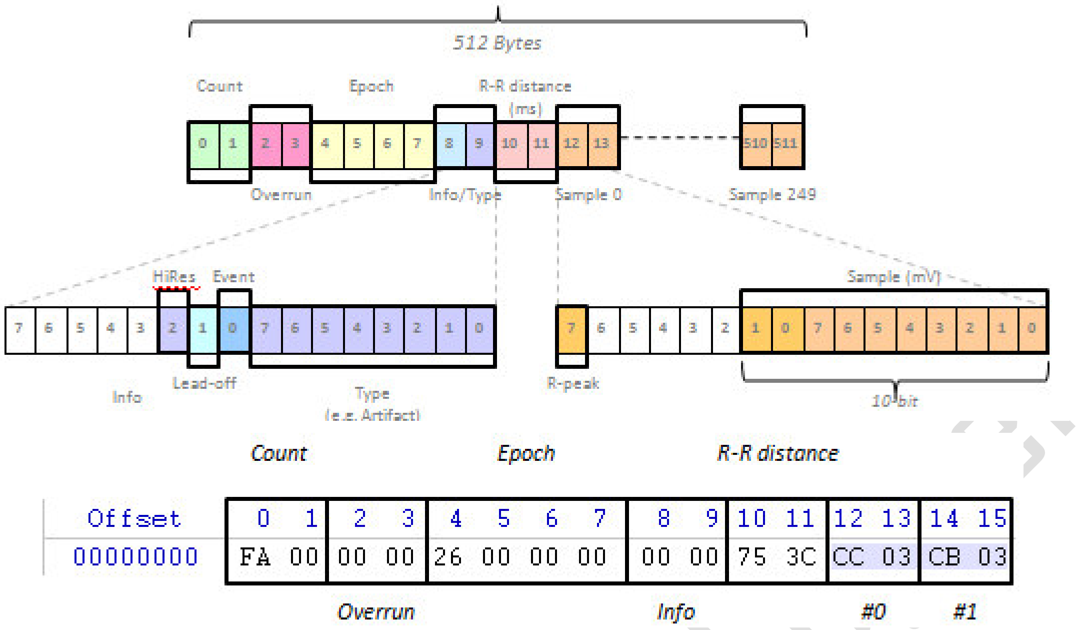


Battery capacity variation (estimation)

HIGH RESOLUTION MODE (10-BIT)

Block values:

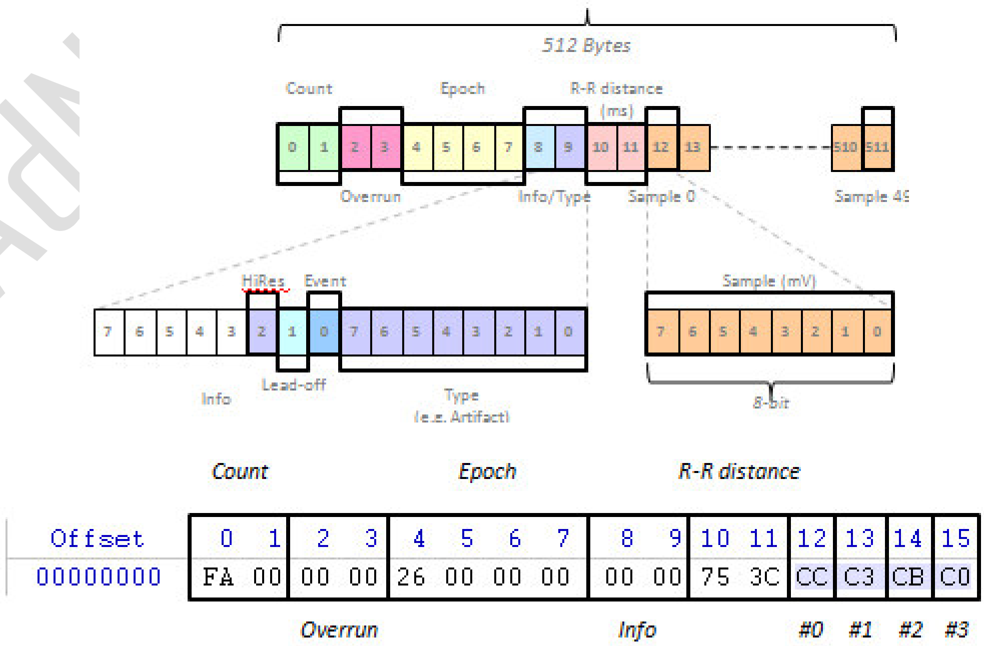
- **Count** Number of samples within the bloc (0-250)
- **Overruns** Number of overflows (missed sampling)
- **Epoch** Number seconds since Jan 01,1970
- **Info**
  - Leads are off
  - EVT button (if pushed in previous second)
  - R-peak type (artifact, normal, ESSV, pause, bradycardia, tachycardia, etc.). Only the last R-peak’s type of block is saved.
- **R-R distance** R-R distance in ms
- **Sample** 10-bit value and bit-7 = 1 (R-peak)



LOW RESOLUTION MODE (8-BIT)

Block values:

- **Count** Nb of samples within the bloc (0-500)
- **Overruns** Nb of overflows (missed sampling)
- **Epoch** Nb seconds since Jan 01,1970
- **Info**
    - Leads are off
    - EVT button (if pushed in previous second)
    - R-peak type (artifact, normal, ESSV, pause, bradycardia, tachycardia, etc.). Only the last R-peak's type of block is saved.
- **Sample** 8-bit value



ECG AND PULSE MODES

In ECG mode the serial outputs the ADC sample values and peaks values are suffixed with additional pieces of information.

- ADC amplitude
- BPM value
- Type class
- R-R distance
- Mean HF value
- Mean BF value
- Min ADC
- Max ADC

These values are very helpful during the debugging sessions for estimating the best values of the algorithm parameters.

PARAMETER'S SETTING

Mean HF threshold is set higher (30+) in the Holter device. This has been done because the ECG Logger can run on noisy "Live ECG signal" when connected to a computer.

Default values:

```
#define MIN_BATTERY    4000        // Minimum voltage level in mV to ensure correct functioning
#define BAT_CONVERT    6.944       // Convert ADC-1 value into mV (battery level)

#define MAX_HF         30          // Maximum HF amplitude for non-artifact ECG signal
                                   (max=1023) Set to 30+ to allow noisy Live ECG signals.
#define MIN_DC         60          // Minimum DC level for non-artifact ECG signal (min=0)
#define MAX_DC         550         // Maximum DC level for non-artifact ECG signal (maw=1023)
#define MIN_ADC        50          // Minimum ADC level for non-artifact ECG signal (min=0)
#define MAX_ADC        1023        // Maximum ADC level for non-artifact ECG signal (strong
                                   signal can overflow so we don't really use it, max=1023)
#define REFRACTORY     68          // Refractory period (>200ms) in nb of samples (68 = 272 ms
                                   or 220 BPM)

#define COND_EXSYS     0.7         // Condition for SV ES (shorter by 30%)
#define COND_PAUSE     2000        // Condition for Pause (longer than 2 secs)
#define COND_BRADY     1091        // Condition for bradycardia (lower than 55 BPM)
#define COND_TACHY     600         // Condition for tachycardia (> 100 BPM for adults)
```

ERROR CODES

Since version 1.6, in order to save place in EEPROM, the error messages are no longer comprehensive. Instead, there are very short error codes displayed.

```
ERR0                               SD card failed!
                                   Check SD, reformat SD card
```

```
ERR1                          Cannot rename temp file
                              Extract the SD and delete the dataxx.hlr file.
                              Clear all files (CLEAR), reformat SD card

ERR2                          createContiguous function failed

ERR3                          contiguousRange function failed

ERR4                          cacheClear function failed

ERR5                          file erase failed

ERR6                          writeBegin function failed

ERR7                          write data sample failed

ERR8                          missed timer event – sampling rate is too high

ERR9                          writeStop function failed

ERR10                         Can't truncate file

ERR11                         Can't rename file

ERR12                         more than 100 files already on SDC!

ERR13                         Parameters string is invalid
                              Check the PARAMS command syntax
```

BILL OF MATERIAL

TOTAL 28,394 €

Supplier	Reference / ASIN	Designation	Quantity	P.U.	Price €
Amazon / TOOGOO	Nano / B00QLOSFDS	Arduino Nano V3.0 AVR ATmega328 P-20AU with cable	1	4,480	4,480
Amazon / Asiawill	AD8232 Heart monitor / B00QNRY4MY	ECG monitor	1	20,110	20,110
Amazon / TOOGOO	CATALEX / B00YMIIL3E	MicroSD Card Adapter shield	1	1,270	1,270
Amazon / SODIAL	032037 / B00QLQQI58	4x battery coupler	1	0,785	0,785
Amazon / Pinzhi	B01CUAX3QK	Black plastic box 10*6*2.5cm	1	1,278	1,278
Amazon	STK0151001508 / B01FJUOEFW	Tact Switch Micro 6x6x16mm	1	0,021	0,021
Amazon / XINTE	$X08382 / B00KNN9KRI	PCB 5x7cm	1	0,45	0,450
		Switch for power	1		
		LED blue 3mm	1		
		10K resistor	1		
		1V5 battery	4		
		Samsung 2GB SDC	1		
		Plastic punch for cover	1		
Amazon / FIAB SpA	F9079 electrodes ECG (43mm x 45mm)	Electrodes	3		

ⓘ NOTE: Other suppliers may offer better prices and quality. Prices may vary due to change rates. The references above have been provided as examples but are not mandatory.

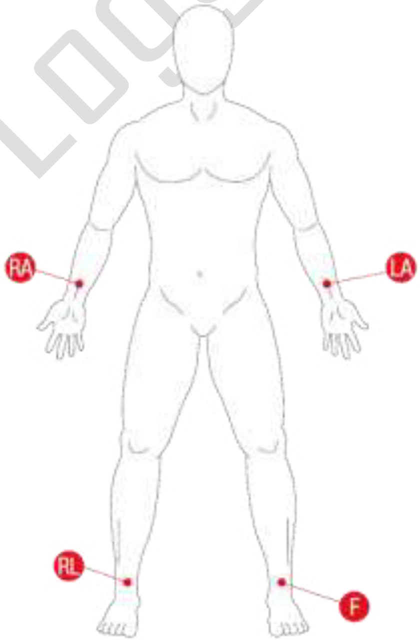
TESTING THE HOLTER DEVICE

NEVER use the device connected to a computer powered by the mains. There is a risk of death.

While testing the hardware and firmware of the Holter device you may want to save the cost of single use electrodes and the pain for placing those on the body. You can use anti-static wrist straps.

Nevertheless, those cannot be used AS IS because they have a 1MΩ resistor in series into theirs cables and these latter must be replaced.

The straps positioning for a good ECG signal is mentioned on the figure below. You may humidify the body parts in contact with the metal part of the strap for a better connection. I use to use the LA/RA/RL positions for the straps. Electrode's foot can be placed independently on the right or left foot.



The other problem to overcome is the noise brought by the USB connection and the 50/60Hz from the mains. Place a ferrite chock on the USB and electrode's cables to reduce high frequency noise. For reducing the mains electrical pollution, you will have to move away your body from electric device and search the best position of the device to minimize the interferences.

For viewing the signal, you can use the Arduino IDE monitor with the "ECG" command but unfortunately the Graph cannot be used because it sends a reset when opened. The ECG Live mode of the ECG Logger Viewer software will be helpful to check the ECG signal waveform.

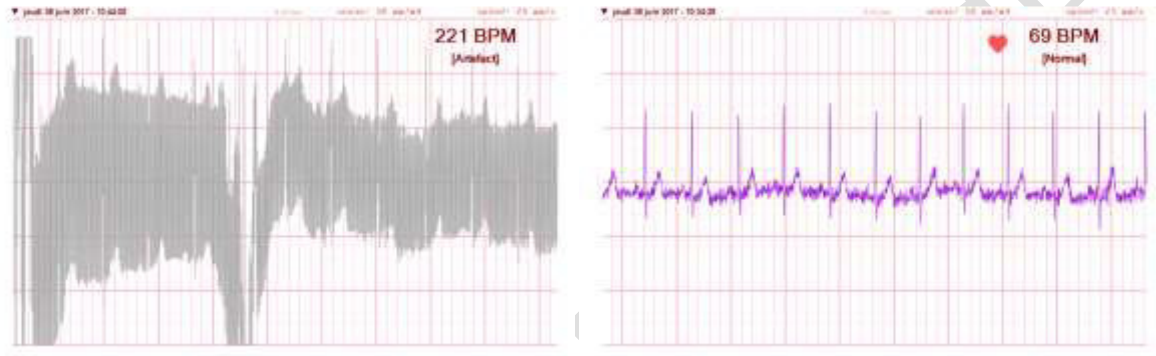
The first time, there will be a fair amount of chance that you get only some awful noise. Don't be afraid! Relax! Leave your arms along your body (muscular noise and body resistivity changes will be reduced); do not make contact between your hands; double check the order of the electrodes; change the position of the device for reducing the noise (most of the noise will disappear when the Holter run standalone).

Improving the ECG signal quality

The two ECG snapshots hereafter have been taken during the same sessions. The first one was shouted right after placing the electrodes and the second after doing some adjustments. We were using the anti-static wrist straps mentioned and made the following operations:

1. One the wrist straps are placed, you must humidify the body's contacts with the electrodes. The foot electrode contact is critical for reducing the level of noise.

2. Move the Holter device, which is connected to the computer to reduce the electromagnetic interferences. Depending on your design and the shielding you have made, you could reduce the noise level by placing your hand over the casing. The USB cable is better be chocked with a ferrite (*see the Package content section*). Move away the device from the computer but long USB cable will not necessary improve the signal quality as it can capture noise too.
3. The cables between the Holter device and the electrodes capture interferences and mainly the mains network ones. You can place a ferrite coil on the cables for reducing the HF noise and keep all cables together and close to your body for limiting the BF noise.
4. Then comes the part of muscular noise and body resistivity part. The only recommendation is "**DO NOT MOVE!**" Any variation of the body resistivity will impair the DC level of the signal which is moving up and down. Any movement, even tiny one like a click on the keyboard will strongly influence the signal.



ECG window snaphots : at STARTUP *AFTER adjustments*

ⓘ *This could discourage you from going further on this project because of the poor results BUT keep in mind that this configuration is the WORST one. It is only to be used for debugging as it facilitates the electrodes placement. Using real electrodes and standalone operation drastically improve the ECG signal quality.*

TROUBLESHOOTING

Problems	Solutions
DEVICE	
Cannot connect the ECG Logger Holter device	• Check in the Peripheral device management that the driver for the Arduino has been correctly installed. • Check if the LED is blinking fast: the SD is not present or correctly inserted. • With a terminal check the message at device start-up. • Check there is no unfinished data transfer still going on (unplug and re-plug the device
Message is "Please check Holter batteries!"	• Check if the power switch is ON. • Check if your batteries are in place and correctly charged. • Select the type of battery inserted into the Holter.
Full charged batteries does not last 24h	• Check if batteries are not too old and correctly charged. • The SD Card used may require too much current. Test with another SD Card brand. • Voltage divider used for battery level measurement is not accurate due to resistors tolerances. *See "Battery Life" section*.
ECG signal is flat	Check the correct plug-in of jack socket.
Events are not recorded	If the ECT button is too quickly depressed, the event is not recorded. But if you depress too long (> 3sec) the Holter will quit the recording mode. A one second press is the best for marking events.
ECG is noisy with lot of artifacts	• Check the quality of the contacts between the electrodes and the skin is crucial. • Do not reuse pads, secure pads with tape for better connection. • When the body is in motion, there are muscular parasitic electrical signals that disturb the ECG signal. • When in Live ECG mode, the Holter is connected to a computer that generates lot of interferences.
LED is continuously quickly blinking	SD card is missing, not formatted or corrupted.
LED is still with short blinks	There is no lead connected or one lead is disconnected
ECG data file download is very long	Yes, the maximum speed rate allowed with the Arduino Nano is about 230 kb/s only. Therefore a 24h ECG file will take 15 minutes to download in LoRes recording mode and 30 minutes in HiRes recording mode.

LICENSES

We are proud to release our ECG Logger under open source licenses. Feel free to use it the way you like in accordance with the licenses below (*no commercial use allowed*).

ECG Logger device HARDWARE	*License : Creative Commons BY-NC-SA* https://creativecommons.org/licenses/by-nc-sa/4.0/
ECG Logger device FIRMWARE	License : GPL v3 (open-source) https://www.gnu.org/licenses/gpl-3.0.en.html

Forum for discussions and downloads is available at https://sourceforge.net/u/adnovea/