ASSEMBLY MANUAL

Contents

Introduction

The ECG device can be built easily and only a basic knowledge of electronics is needed to realize the hardware circuit. No software programming knowledge is required since all you need is to install the App by opening the apk file from an Android smartphone and to upload the provided Arduino sketch on the Arduino board (this can be done easily by using the Arduino Software IDE and one of the many tutorials available on the web). However, source files are available to modify or personalize the App (Android programming skills needed) and/or Arduino program (Arduino programming skills needed).

The ECG device is intended only as a design research project and it is NOT a medical device, so please read the Warnings and safety issues before going on.

Warnings and safety issues

The ECG device is intended only as a design research project and it is NOT a medical device.

Use ONLY battery (max voltage supply: 9V).

DO NOT use any AC power supply, any transformer or any other voltage supply to avoid serious injury and electrical shock to yourself or others. Do not connect any AC-line powered instrumentation or device to the ECG device here proposed.

The ECG device is electrically connected to a person and only low voltage batteries (max 9V) must be used for safety precautions and to prevent damage to the device.

Placement of the electrodes on the body provides an excellent path for current flow. When the body is connected to any electronic device, you must be very careful since it can cause a serious and even fatal electric shock.

The authors cannot be responsible for any harm caused by using any of the circuits or procedures described in this manual. The authors do not claim any of the circuits or procedures are safe. Use at your own risk. It is imperative that anyone who wants to build this device has a good understanding of using electricity in a safe and controlled manner.

Hardware description

The simple circuit design and layout are a good compromise for having both a low cost (few components) and good performance.

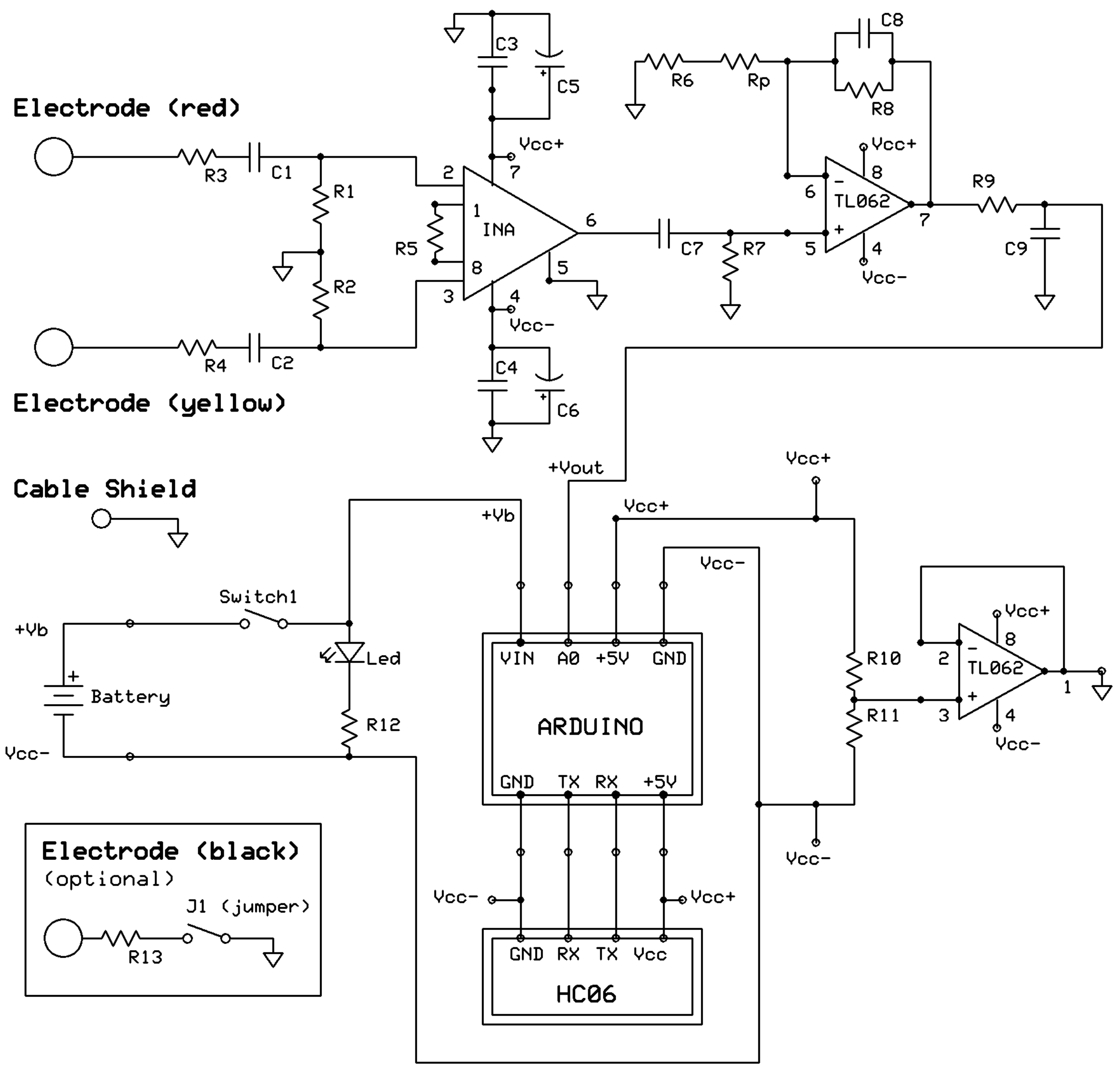


Figure 1: ECG hardware circuit

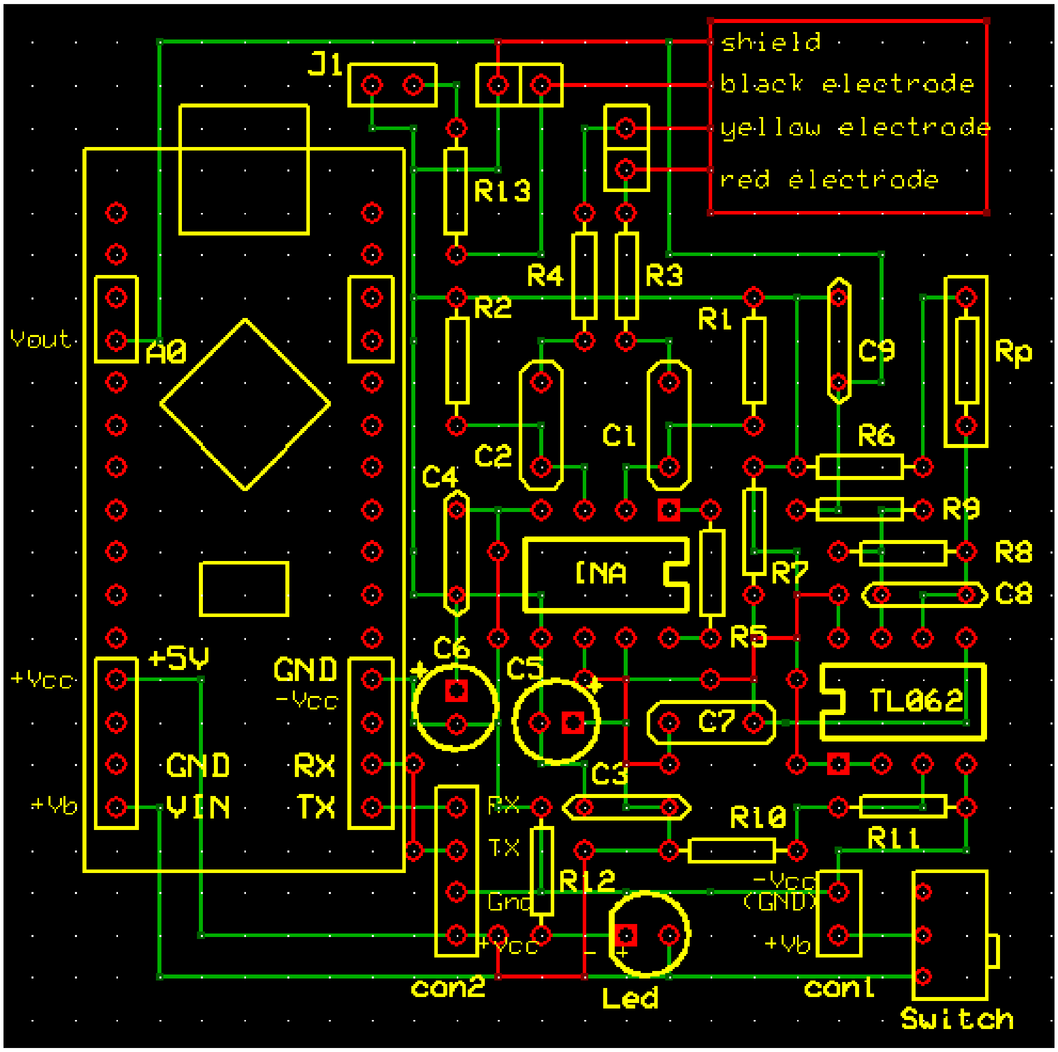


Figure 2: PCB layout: component (yellow), top copper layer (red), bottom copper layer (green)

The battery supplies (+Vb) the Arduino board and the led L1 when the device is turned on; the rest of the device is supplied by the Arduino 5 V voltage output (+Vcc). Basically the device works between 0 V (-Vcc) and 5 V (+Vcc), however the single supply is converted to dual supply by a voltage divider with equal resistors (R10 and R11), followed by a unity gain buffer (1/2 TL062). The output has 2.5 V (the mid-voltage of the TL062 power supply: 0-5 V); the positive and the negative rails of power then give a dual supply (± 2.5 V) with respect to the common terminal (reference value).
The capacitors C3, C4, C5 and C6 make the voltage supply more stable.

For safety issue, each electrode is connected to the device through a protection resistor of 560 kΩ (R3, R4, R13) to limit the current flowing into the patient in case of a fault inside the device. These high resistors (R3, R4, R13) should be used against the rare situation when the low

voltage power (6 or 9 V, according to the used battery supply voltage) comes directly to the patient leads accidentally, or due to the INA component failing.
Besides, two CR high pass filters (C1-R1 and C2-R2), placed at two inputs, block the dc current and reduce unwanted dc and low frequency noise generated by contact potentials of electrodes. ECG signal is so high pass filtered before the amplifying stage with a cut off frequency around 0.1 Hz (at -3 dB). The presence of R1 (as R2) reduces the input impendence of the pre-amplification stage so that the signal is reduced by a factor depending on the value of the R1 and R3 (as R2 and R4); such factor can be approximated as:

$$R1 / (R1 + R3)$$ [1]

It is more advisable to choose the couple C1 - C2 with capacity values very close each other, the couple R1- R2 with resistance values very close each other and the same for the couple R3 - R4. In this way, an unwanted offset is reduced and not amplified by the instrumentation amplifier (INA128). Any mismatch between circuit parameters of the components in the dual input circuit contributes to a degradation of the CMRR; such components should be very well matched (even the physical layout) so that their tolerance should be chosen as low as possible (alternatively the operator can measure their values manually with a multimeter in order to choose the couple components with the values as close as possible).

R5 defines the INA128 gain according to the formula:

$$G_{INA} = 1 + (50\ k\Omega / R5)$$

ECG signal is so amplified by the INA and successively high pass filtered by C7 and R7 (with a -3 dB cut off frequency around 0.1 Hz) to eliminate any dc offset voltage before the last and higher amplification made by the operation amplifier (1/2 TL062) in a non-inverting configuration with a gain:

$$G_{TL062} = 1 + (R8 / (Rp+R6))$$

To let the user change the gain at runtime, the operator can choose to use a variable resistor (trimmer / potentiometer) instead of Rp or a female socket strip for a resistor that can be changeable (because not soldered). However, in the first case it is not possible to know exactly the actually gain of the ECG signal (the values in mV of the data will be not correct) while in the second case it is possible to have the correct values in mV by specifying the value of Rp in the formula “Gain” inside the “Setting” page of the app (see “User Manual” file).
C8 capacitor creates a low pass filter with a -3 dB cut off frequency around 40 Hz as the RC filter composed by R9 and C9.
The cut off frequency value is given by the formula:

$$f = 1 / (2*\pi*C*R).$$

---

[1] Equal to 0.797 if R1 = 2.2 MΩ and R2 = 560 kΩ

For low pass filters @ 40 Hz [2], RC components values are:

R8 = 120 kΩ
C8 = 33 nF
R9 = 39 kΩ
C9 = 100 nF

The ECG signal is so filtered in a band between 0.1 and 40 Hz and amplified with a gain equal to:

$$Gain = 0.797 * G_{INA} * G_{TL062}$$

Since R5 = 2200 Ω , R8=120000 Ω, R6=100 Ω, Rp=2200 Ω ,

$$Gain = 0.797 * (1+50000 / 2200) * (1+120000 / (2200 + 100)) = 1005$$ [3]

To have accurate values for the filter cut-off frequencies, RC filter components should have a tolerance as low as possible (alternatively the operator can measure their values manually with a multimeter in order to choose the ones closest to the desired value).

The analog signal is digitalized by the Arduino board (A0 input channel) and then transmitted to the HC-06 module by the serial communication pins; finally, the data are sent to the smartphone by Bluetooth.
The reference electrode (black) is optional and can be excluded by removing the jumper J1 (or the operator can use a switch instead of the jumper). The circuit configuration is designed to work also with two electrodes; however, the reference electrode should be used to have a better signal quality (lower noise).

---

[2] The upper band limit of the low pass filter can be increased by changing R8 or C8 and R9 or C9. For a Frequency Bandwidth @ - 3dB (Hardware) = 0.1 Hz - 150 Hz , RC filter components must be changed according to the following values:

R8 = 47000 Ω
C8 = 22 nF
R9 = 47000 Ω
C9 = 22 nF

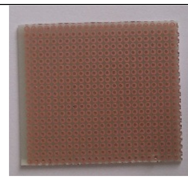
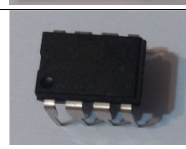
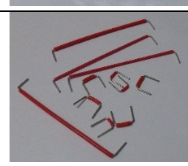
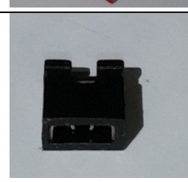
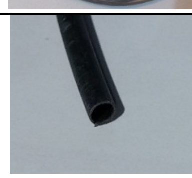
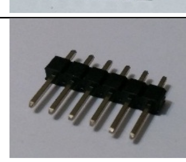
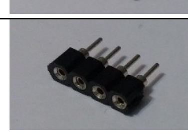
In this case, the ECG SmartApp version for bandwidth at 150 Hz should be used since further digital low pass filters (at 100 Hz and 150 Hz) are available.

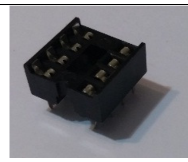
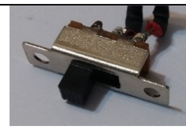
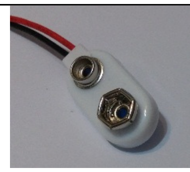
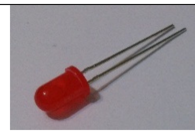
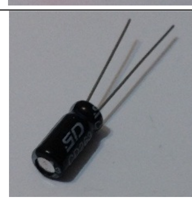
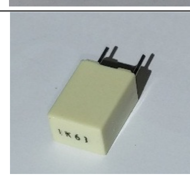
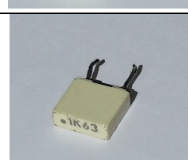
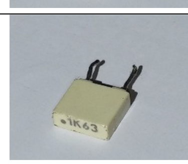
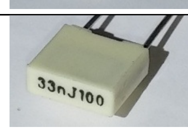
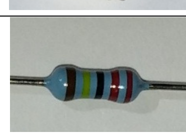
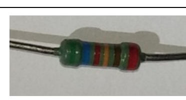
[3] In case of ECG SmartApp version for bandwidth at 150 Hz, R5 = 2200 Ω , R8=47000 Ω, R6=220 Ω, Rp=680 Ω, then:

$$Gain = 0.797 * (1+50000 / 2200) * (1+47000 / (680 + 220)) = 1006$$

Components

By excluding the Smartphone and disposable parts (electrodes and batteries), the whole device cost is around 43 US dollars (here considered the single product; in case of a larger quantity, the price would go down).

Component description	Picture	Quantity	Unit price [4] [US dollars]	Cost [US dollars]	Circuit element
Perforated prototype board (dimensions: 23x21 holes - around 60 mm x 60 mm)		1	4	4	
INA128P		1	6	6	INA128
Cable (spare parts to connect components on the board)		1 meter		0.20	
Jumper connectors		1	0.1	0.1	J1
Coaxial cable (to connect electrodes to the board)		3 meters		3	
Heat Shrink Tubing		0.5 meter		0.5	
Female header connectors (more parts with at least 22 number of positions)				1	
Male header connectors (more parts with at least 14 number of positions)				1	
Female Socket Strip - 1 Row Straight Through Hole (4 number of positions)				0.2	For Rp

Sockets DIP (8 positions)		2	0.2	0.4	For INA128 and TL062
Slide Switch		1	0.4	0.4	S1
Battery strap [5]		1	0.6	0.6	
Led - Through Hole – red		1	0.3	0.3	L1
Electrolytic capacitors: 1 μF (low voltage)		2	0.1	0.2	C5, C6
Film capacitors: 1 μF (low voltage)		3	0.1	0.3	C1, C2, C7
Film capacitors: 100 nF (low voltage)		2	0.1	0.2	C3, C4
Film capacitor: 100 nF (low voltage) [6]		1	0.1	0.1	C9
Film capacitor: 33 nF (low voltage) [7]		1	0.1	0.1	C8
Resistors: 2.2 MΩ (Through Hole, 0.25 W , tolerance 5% or 1%)		3	0.05	0.15	R1, R2, R7
Resistors: 1 MΩ (Through Hole, 0.25 W , tolerance 5% or 1%)		2	0.05	0.1	R10, R11
Resistors: 560 kΩ (Through Hole, 0.25 W , tolerance 5% or 1%)		3	0.05	0.15	R3, R4, R13

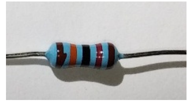
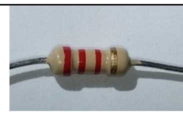
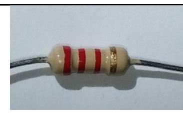
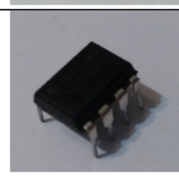
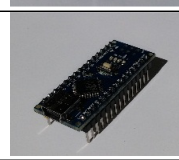
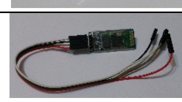
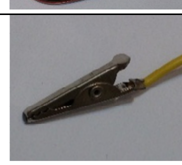
Resistor: 120 kΩ (Through Hole, 0.25 W , tolerance 5% or 1%) [8]		1	0.05	0.05	R8
Resistor: 39 kΩ (Through Hole, 0.25 W , tolerance 5% or 1%) [9]		1	0.05	0.05	R9
Resistor: 10 kΩ (Through Hole, 0.25 W , tolerance 5% or 1%)		1	0.05	0.05	R12
Resistor: 2.2 kΩ (Through Hole, 0.25 W , tolerance 5% or 1%)		1	0.05	0.05	R5
Resistor: 2.2 kΩ (Through Hole, 0.25 W , tolerance 5% or 1%) [10]		1	0.05	0.05	Rp
Resistor: 100 Ω (Through Hole, 0.25 W , tolerance 5% or 1%) [11]		1	0.05	0.05	R6
TL062CP		1	0.7	0.7	TL062
Arduino/Genuino board (with male header connectors) [12]		1	15	15	Arduino
HC-06 bluetooth module + cable		1	6	6	HC06
Alligator clip		3	0.2	0.6	
Box		1	2	2	
TOT				**43.6**	

Table 1: components list

[4] Approximate price based on the catalogs of some large electronic component distributors of North America and Europe.

[5] Battery holder may be needed in case of 6 x 1.5 V AA batteries (cost around 1 dollar) or a 6V voltage supply with 4 x 1.5 V AA batteries (cost around 1 dollar)

[6] C9 = 22 nF in case of ECG SmartApp version for bandwidth at 150 Hz

[7] C8 = 22 nF in case of ECG SmartApp version for bandwidth at 150 Hz

[8] R8 = 47 kΩ in case of ECG SmartApp version for bandwidth at 150 Hz

[9] R9 = 47 kΩ in case of ECG SmartApp version for bandwidth at 150 Hz

[10] Rp = 0.68 Ω in case of ECG SmartApp version for bandwidth at 150 Hz

[11] R6 = 0.22 Ω in case of ECG SmartApp version for bandwidth at 150 Hz

[12] Price can vary according to the used Arduino board (Nano, UNO, Micro, etc)

Needed Tools

Tester, clippers, soldering iron, solder wire, screwdriver and pliers.

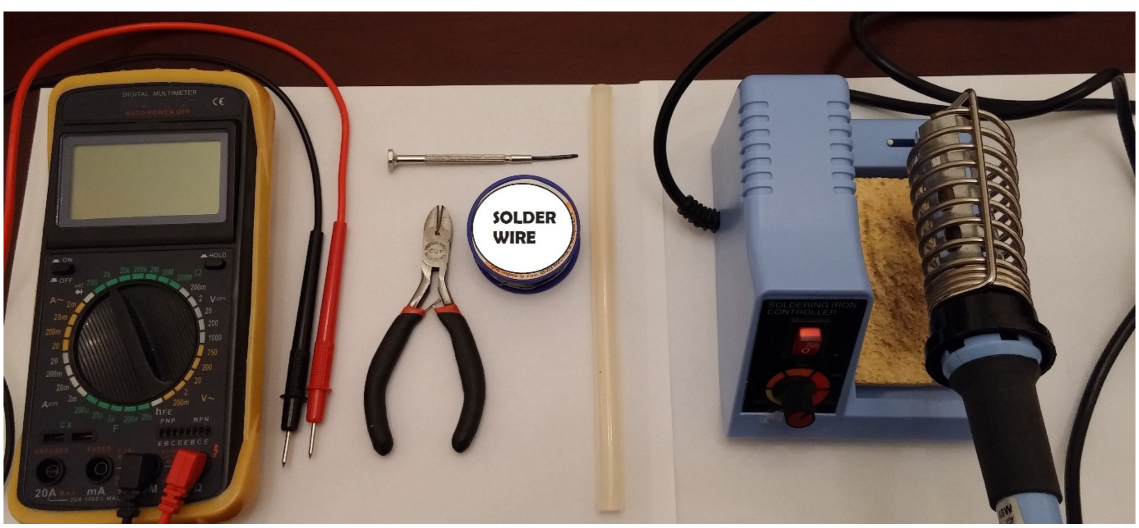


Figure 3: tools

How to build

- Prepare a perforated prototype board with 23x21 holes (around 62 mm x 55 mm)

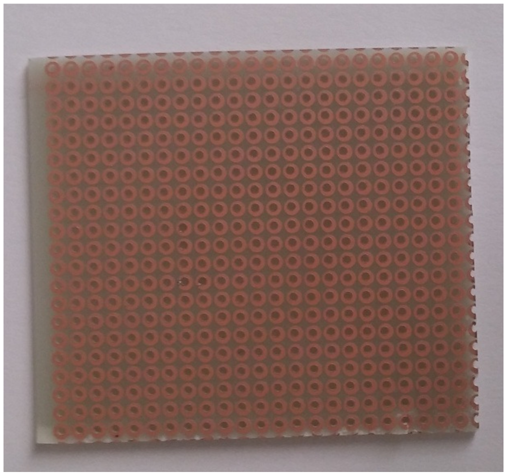


Figure 4: perforated prototype board

- According to the PCB top layout (figures 6 and 7), solder:
    - resistors
    - connecting wires (red lines in figure 5)
    - female socket strip (for Rp)
    - sockets
    - male and female header connectors (female header connectors position here reported in the figures is suitable for Arduino Nano or Arduino Micro)
    - capacitors
    - Led

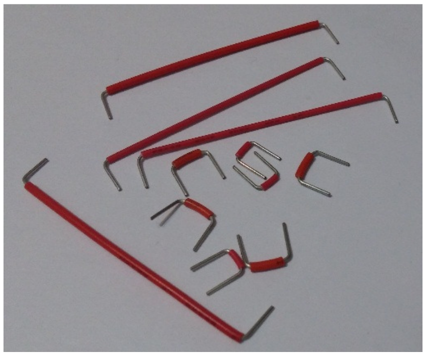


Figure 5: connecting wires

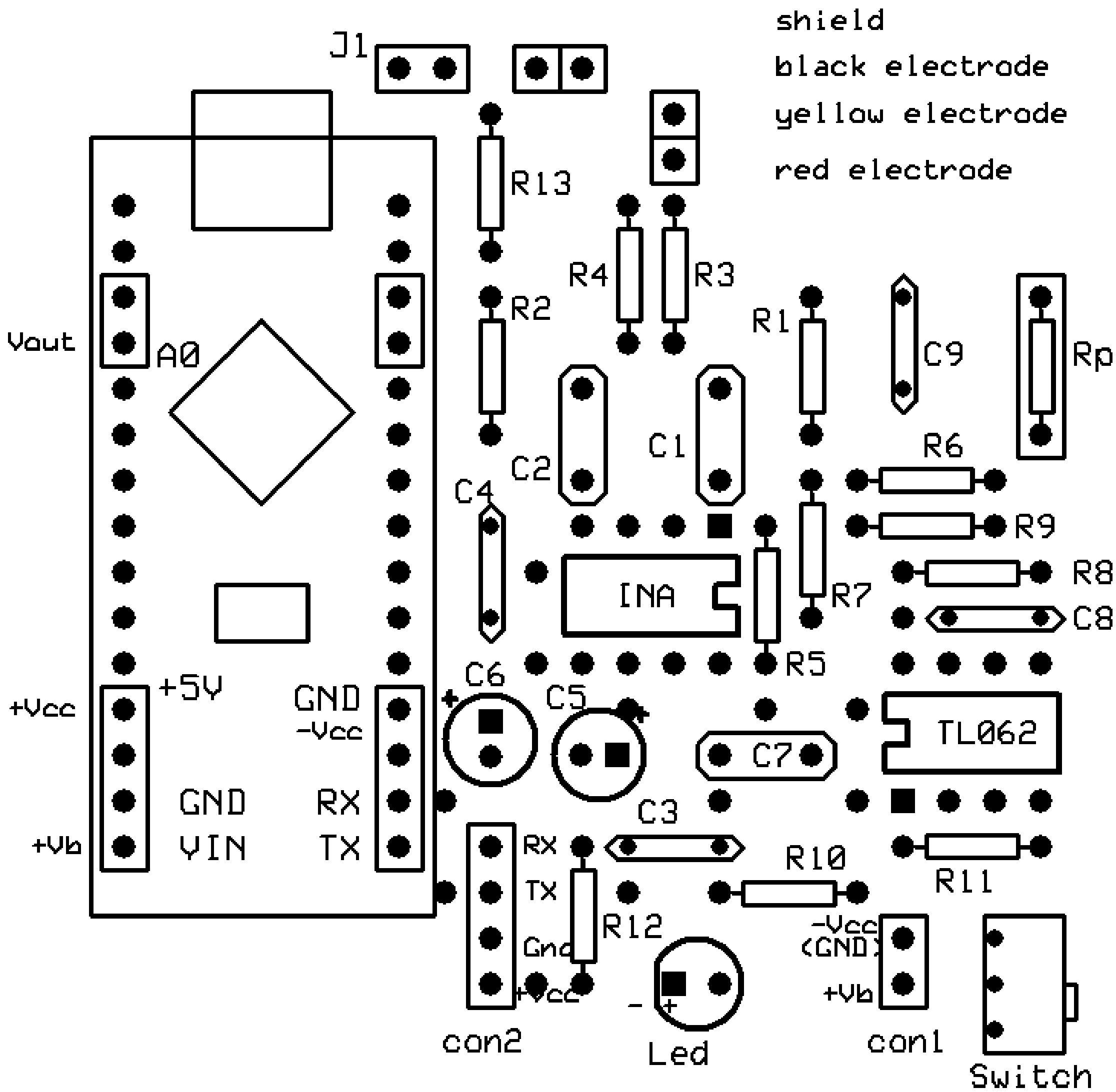


Figure 6: PCB components layout (top view)

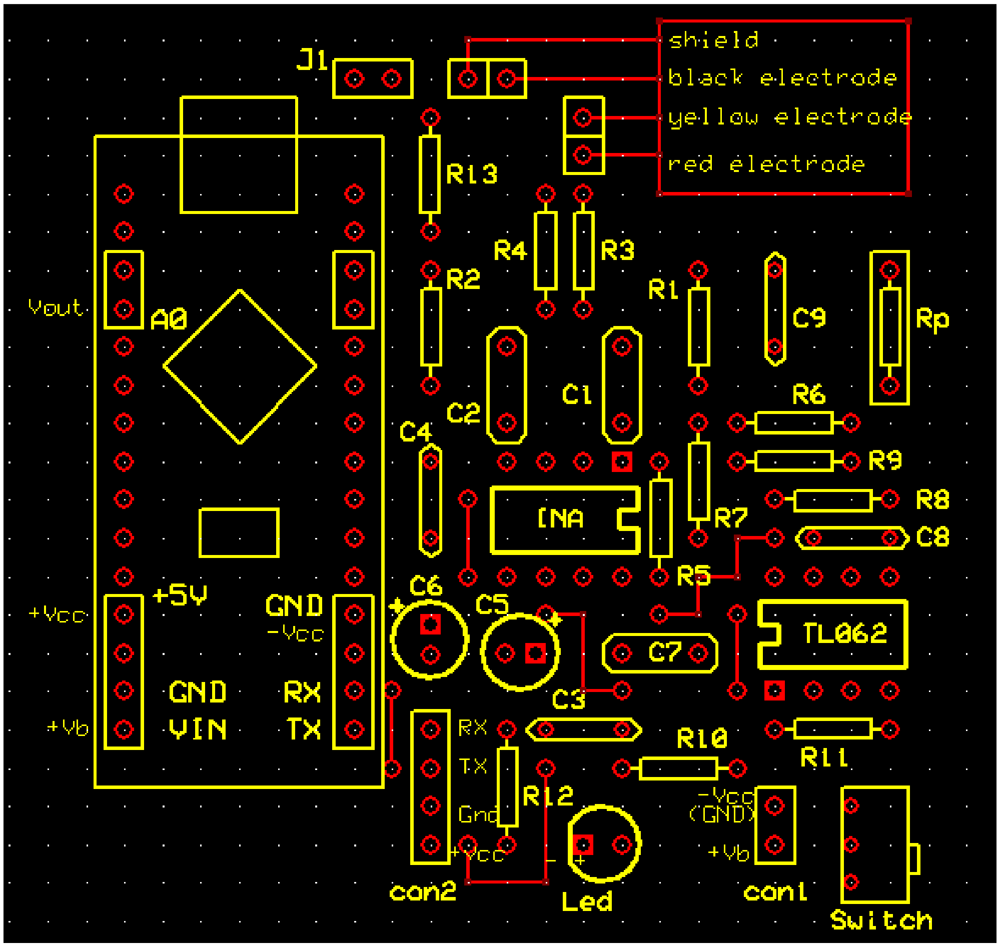


Figure 7: PCB top view with components (yellow) and connecting wires (red)

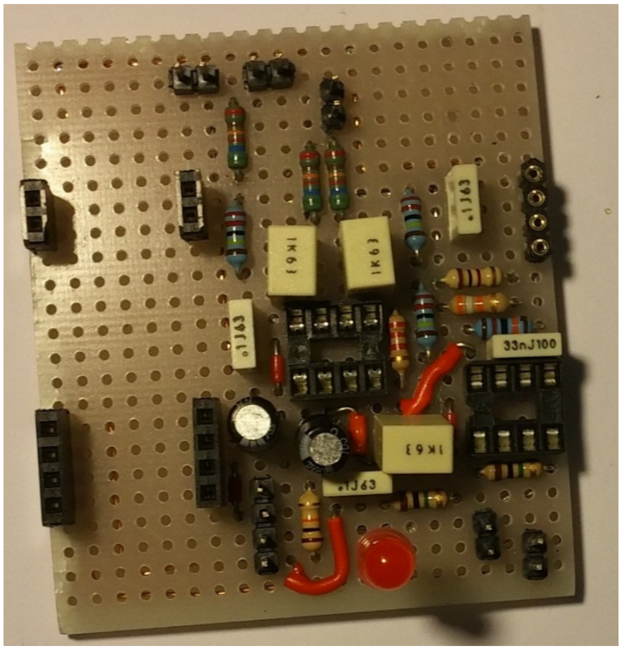


Figure 8: circuit top view

- Connect all components according to the PCB bottom layout (figure 9)

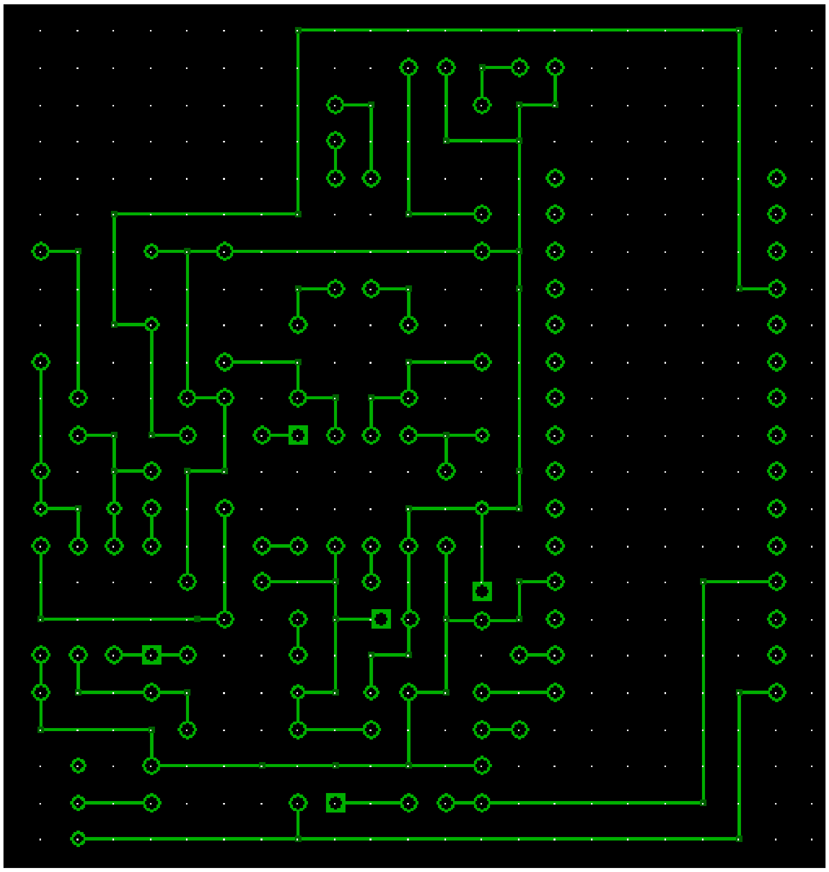


Figure 9: PCB bottom view

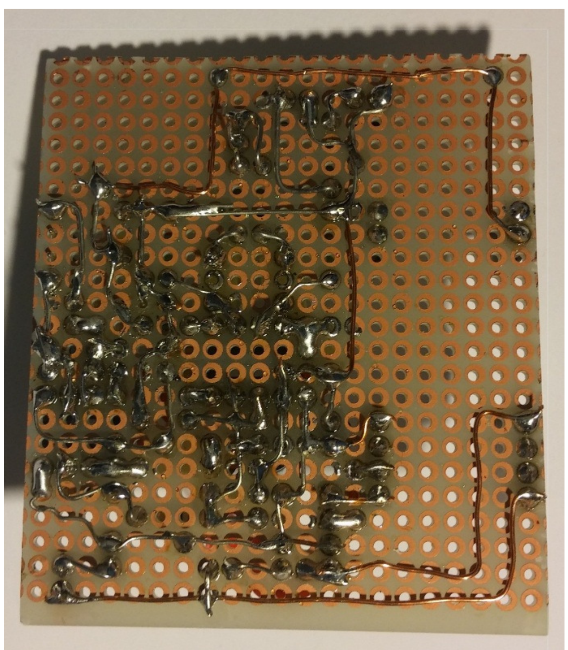
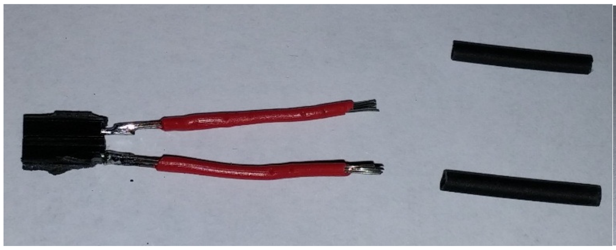
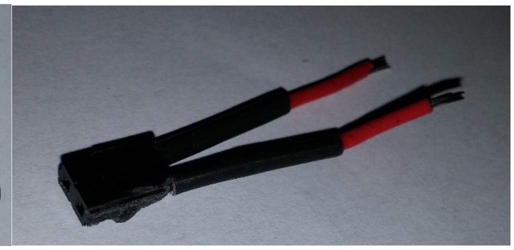


Figure 10: circuit bottom view

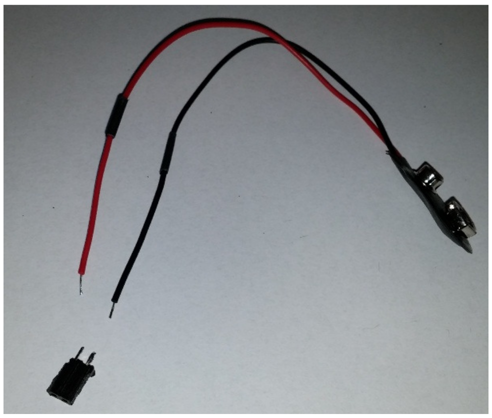
- Realize a wire connector for the battery as shown in figure 11 (a, b and c) using the battery strap/holder, female header connectors and heat shrink tubing; connect it to the PCB “con1” (connector1)



a)



b)



c)

Figure 11: battery connector

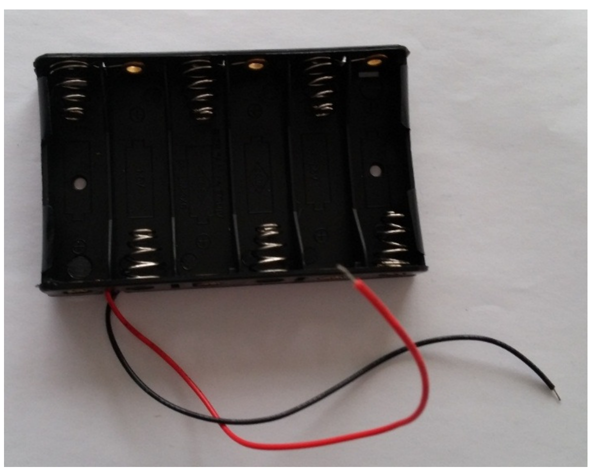


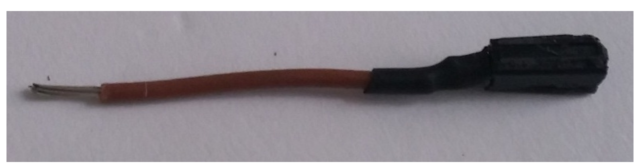
Figure 12: Battery holder may be needed in case of 6 x 1.5 V AA batteries (or a 6V voltage supply with 4 x 1.5 V AA batteries)

- Realize three electrodes cables as shown in figures 13 and 14 (using the coaxial cable, female header connectors, heat shrink tubing, alligator clip) and connect them to the PCB tightening them to the board with some rigid cables (figure 15)

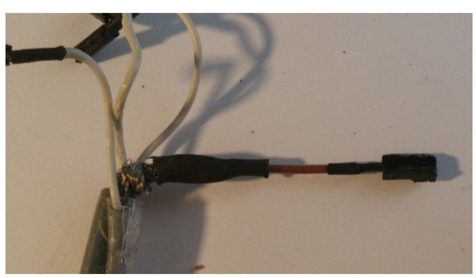

a) b)


c) d)

Figure 13: cable end to be connected to the electrodes

a) b)



c)

Figure 14: cable end to be connected to the PCB

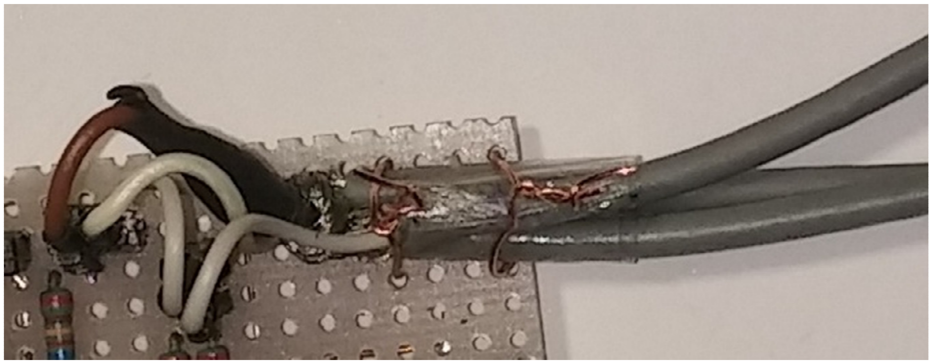
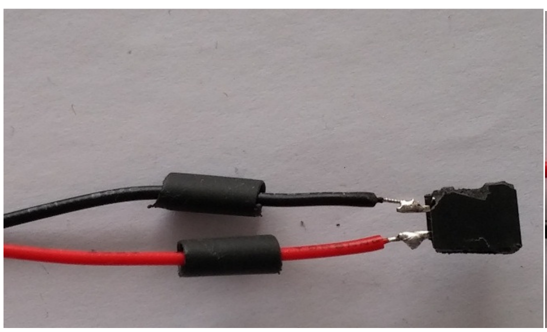
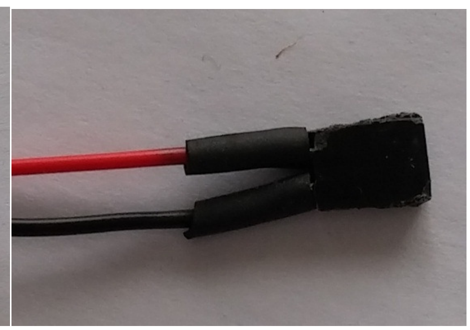


Figure 15: connect the cables to the PCB tightening them to the board with some rigid cables

- Realize a switch as shown in figure 16 (using the slide switch, female header connectors, heat shrink tubing) and connect it to the PCB

a) b)

c)

Figure 15: the switch can be also soldered directly to the PCB without using any connector

- Place the INA128, TL062 and Rp resistor into the correspondent sockets

- Program (see “Software Description” paragraph) and connect the Arduino Nano board (perforated prototype board and female header connectors should be adjusted on the PCB if another Arduino board (e.g. UNO or Nano) is used)

- Connect the HC-06 module to PCB “con2” (connector2)

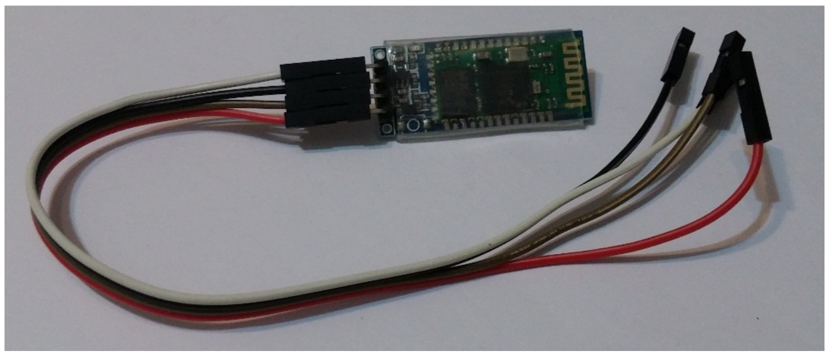


Figure 16: HC-06 module and its cable

- Connect the jumper J1 to use the reference electrode

- Connect the battery

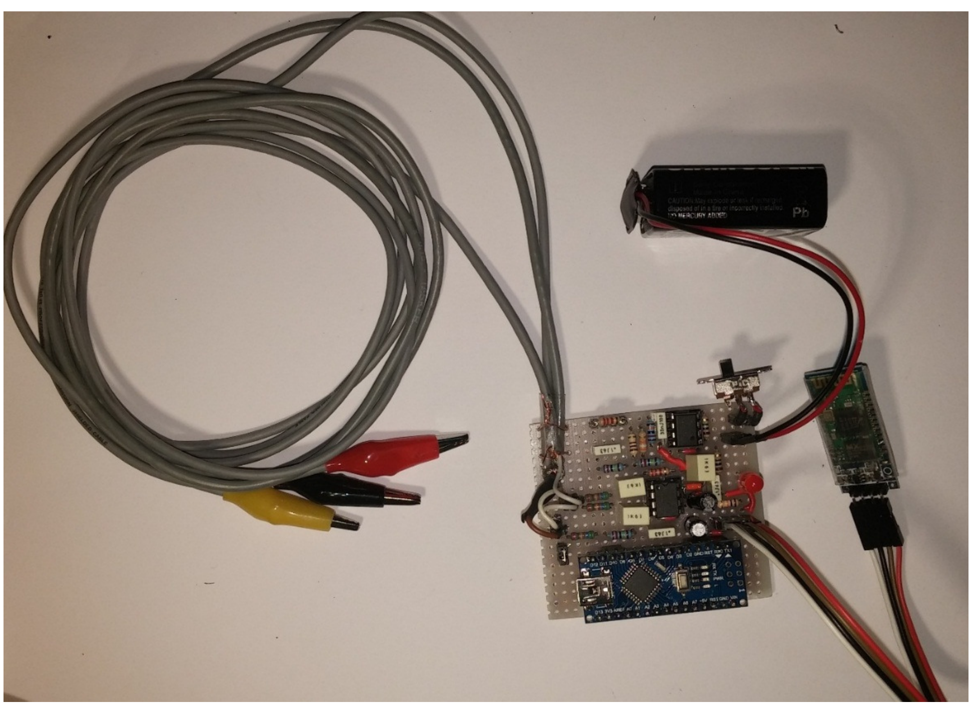
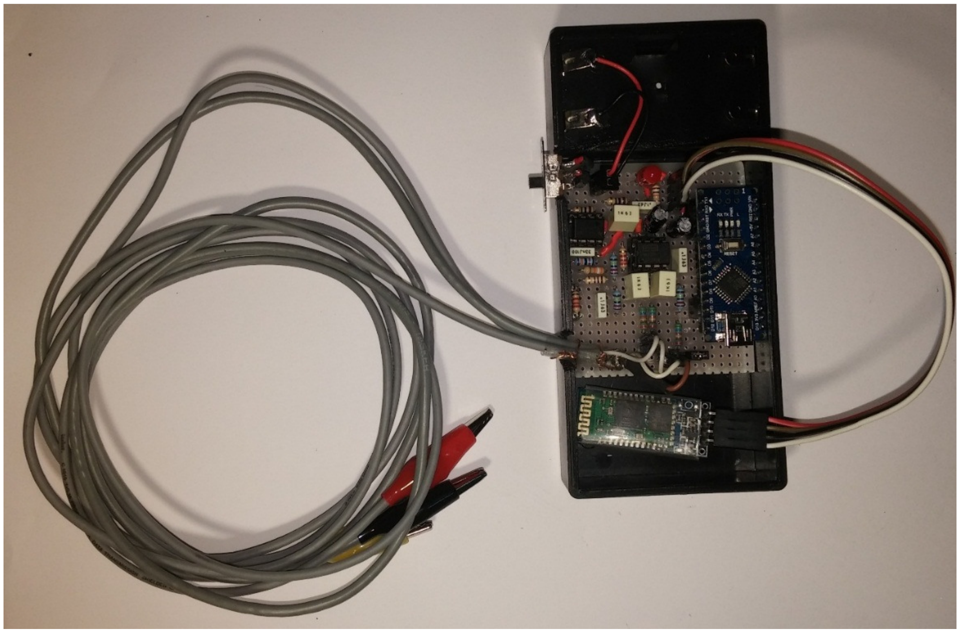


Figure 17: ECG device hardware completed

- Place the circuit inside a suitable box with holes for the Led, the cables and the switch.



a)

b)

Figure 18: ECG device inside a suitable plastic box

Other options

- The ECG signal for monitoring application is filtered between 0.1 and 40 Hz; the upper band limit of the low pass filter can be increased by changing R8 or C8 and R9 or C9 (see table 1).

- Instead of the Rp resistor, a trimmer or potentiometer can be used to change the gain (and amplify the ECG signal) at runtime.

- The ECG device can work also with different Arduino boards. Arduino Nano and Arduino UNO were tested. Other boards can be used (such as Arduino Micro, Arduino Mega, etc.) however the provided Arduino sketch file needs modifications according to the board features.

- The ECG device can work also with the HC-05 module instead of HC-06 one.

Software Description

No software programming knowledge is required.

- **Arduino Programming:** Arduino sketch files can be uploaded on the Arduino board easily by installing the Arduino Software IDE (free download from the Arduino official web site) and following the tutorial available on Arduino official website. A single sketch file ("ECG_SmartApp_skecht_arduino.ino") for both Arduino Nano and Arduino UNO is provided (the sketch was tested with both the boards). The same sketch should work also with Arduino Micro (this board was not tested). For other Arduino board, the sketch file may need changes.

- **Installing the ECG SmartApp:** To install the App, copy the provided apk file "ECG_SmartApp.apk" (or "ECG_SmartApp_upTo150Hz.apk" in case of the version for bandwidth at 150 Hz) on the smartphone memory, open it and follow the instruction by accepting the permissions. Before installing, it may be needed to change the smartphone setting by allowing installation of app from unknown sources (tick the box of "Unknown sources" option of the "Security" menu). To connect the ECG device with the HC-06 (or HC-05) Bluetooth Module, pairing code or password may be asked in case of the first Bluetooth connection with the module: enter "1234". If the App does not find the Bluetooth Module, try to pair the smartphone with the HC-06 (or HC-05) Bluetooth Module by using the smartphone Bluetooth Setting (pairing code "1234"); this operation is needed only once (first connection)

Android Source Files

Source files are available to modify or personalize the App. However, Android programming skills are needed.
The archive “Android_Source_Files.zip” contains all useful files to be added to the app project (“Android_Source_Files_upTo150Hz.zip” in case of the version for bandwidth at 150 Hz).

The app can be built by creating a new android project containing the following files:

- Activities:
    - MainActivity.java (the main activity to record ECG signal)
    - Pag1.java (the setting page)
    - Pag2.java (to open and visualize ECG files)

- Manifest:
    - AndroidManifest.xml

- App icon (Drawable):
    - app_icon.png

- Resources:
    - Layouts:
        - activity_main.xml
        - pag1.xml
        - pag2.xml
    - Menu:
        - menu_filter.xml
        - menu_filter_2.xml
        - menu_leads.xml
        - menu_notch.xml
        - menu_savedatafiltered.xml
    - Raw:
        - example1.txt (ECG example files)
        - example2.txt (ECG example files)
        - example3.txt (ECG example files)
        - help.pdf (User Manual guide)
    - Values:
        - Dimens.xml
        - String.xml
        - Styles.xml

- Libraries:
    - achertengine-1.1.0.jar
    - android-support-v4.jar

The app should include the "AChartEngine" (achertengine-1.1.0.jar) a charting software library for Android applications. It is an open-source framework, free to use in any free or commercial application (free download available on http://www.achartengine.org/)
About the Target Android Devices, choose "Google APIs 3.1" and "API level 12":

- Minimum required SDK: API 12
- Target SDK: API 12
- Compatible with: Google APIs (API 12)

The app can be build using software as Eclipse or Android Studio (the official integrated development environment for the Android platform).

Future development

To realize a new version that makes possible the ECG recording of all conventional leads (not only the limb leads, but also the augmented limb leads and the precordial leads).