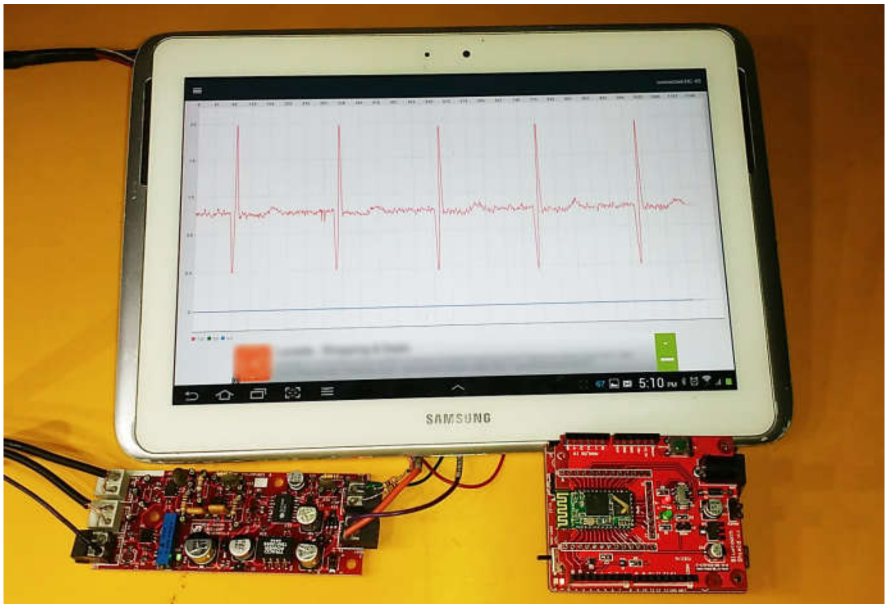
ECG Monitor Using gizDuino and Bio-Amp

Application Reference



Materials Needed:

- Bio-Amp Module
- gizDuino Plus/328
- Bluetooth Shield
- Android Phone/Tablet
- 12V Power Supply
- ECG Electrodes
- A friend willing to be your practice subject

1. Install the Bluetooth Terminal/Graphics app to your Android phone/tablet. The link to the mentioned app is provided below for your convenience.

https://play.google.com/store/apps/details?id=com.emrctn.BluetoothGraphics&hl=en

2. Run the app and prepare it for the ECG project by checking on Settings>Graph Values Setting>Refreshing Graphs.

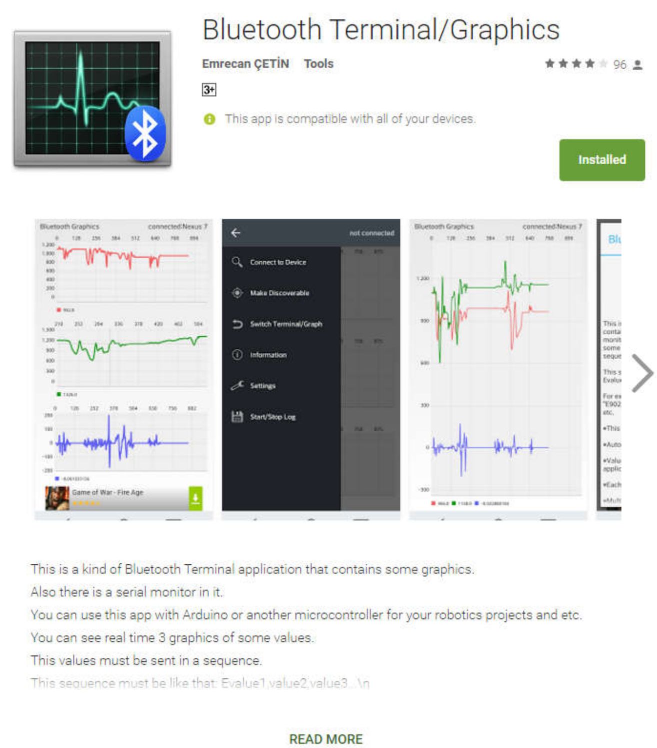


Figure 1. Bluetooth Terminal/Graphics app display data from your Bluetooth ECG setup very much like a digital oscilloscope using a simple bluetooth data trasfer protocol.

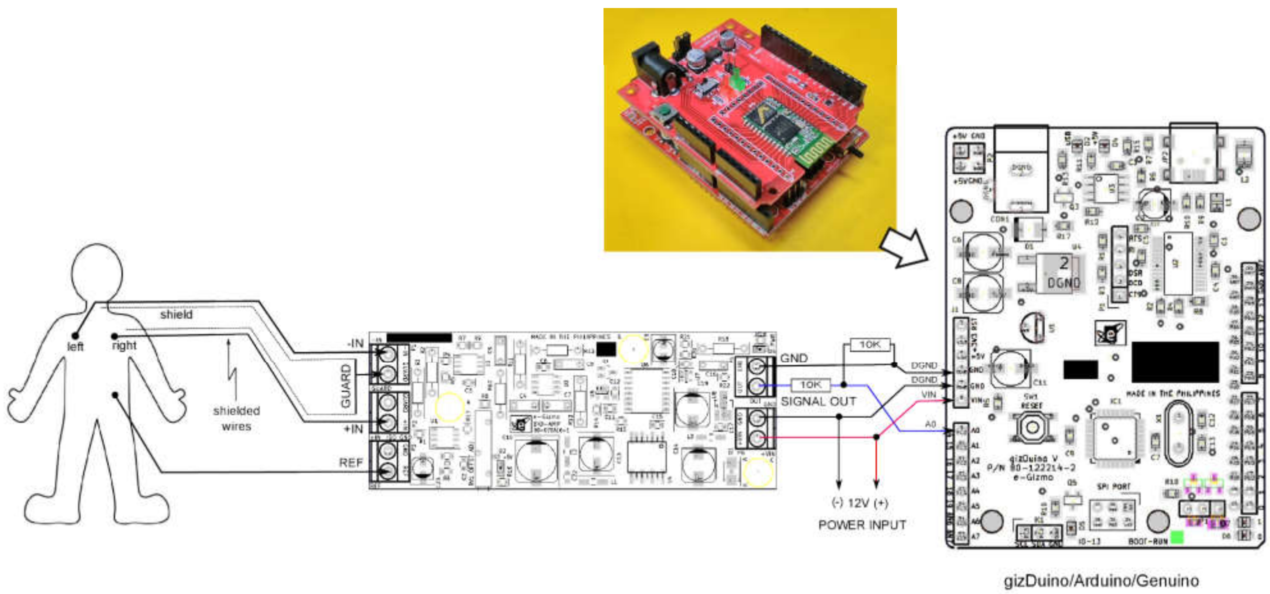
3. Download and load the ECG Bio-Amp sketch provided in the Bio-Amp product page website to your gizDuino/Arduino board.

4. Set up your Bluetooth Shield to slave mode and 115200bps baud rate. Please refer to the product documentation of the bluetooth shield for detailed discussion on how to do this.

5. Wire the Bio-Amp with your gizDuino as shown in figure 2.

6. Install the Bluetooth shield and power up the system with a stable 12V DC source.

7. Discover and pair the ECG Bluetooth with your Android device. This is usually done through your Android's Setting apps. Look for a bluetooth device labeled as "HC-05", and pair it with your phone. The default password is "1234".



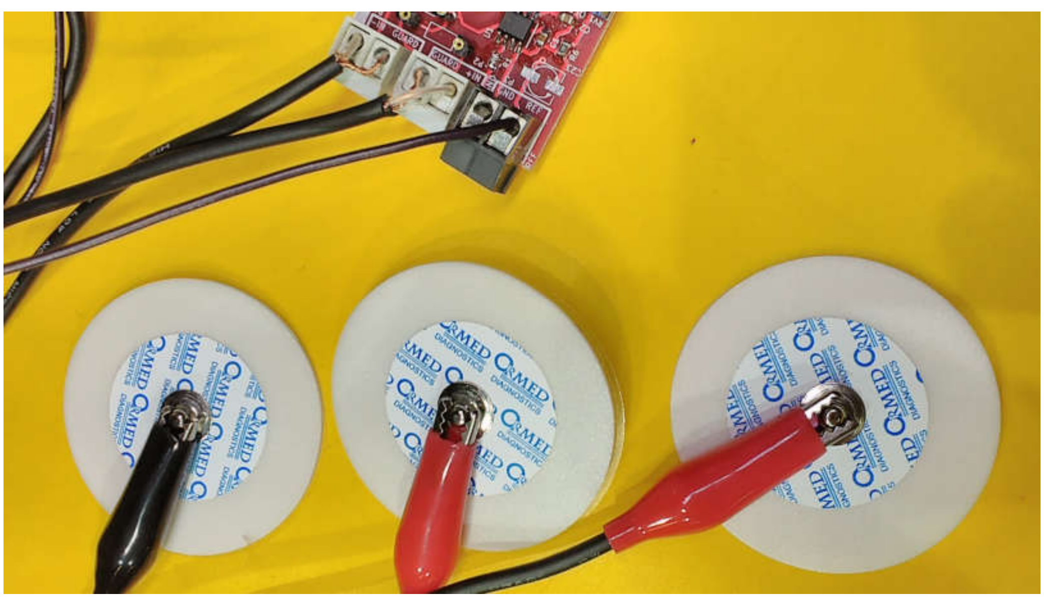


Figure 3. In our experimental setup, we used shielded wires connected as presented in figure 2. Alligator clips were used to connect with the electrode pads.

8. Launch the Bluetooth Terminal/Graphics app and connect with the ECG device paired in step 7.

9. Connect the ECG electrodes to your friend as diagrammed in figure 2. You should see an ECG waveform appearing on your Android Device by now. If you are seeing just a straight line, check first to ensure your friend is alive and well. And then gently adjust RV1 until an ECG waveform appears. Continue adjusting to the desired 0V reference. Make sure signal peaks are not clipped. If clipping occurs, you may have to shift to 0V reference or reduce the gain of the Bio-Amp. See Bio-Amp reference manual for more details.

Figure 4. You will probably see your ECG setup displaying something like this when you run it with the electrodes un-attached.

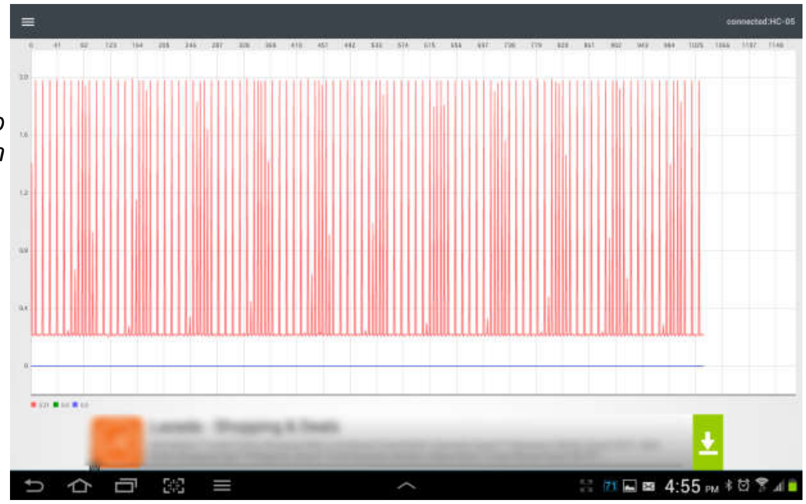


Figure 5. An ECG output from the ECG project showing clipping at the low side peaks.

Figure 6. Clipping occuring in Fig.5 is eliminated by adjusting RV1.

```
float data1=1.25;

#define RATE 3      // specify sampling rate in ms. Set to 3ms in this example
#define SR 62.5*RATE


void setup() {

  Serial.begin(115200);   // Set up Serial for 115.2 kbaud speed

  //Timer 1
  // Timer 1 interrupt service is used exclusively for data acquisition in order
  // to achieve a uniform sample rate
  // That means you cannot not use functions and libraries that also uses timer1
  // like the servo library

   // initialize timer1
  noInterrupts();           // disable all interrupts
  TCCR1A = 0;
  TCCR1B = 0;
  TCNT1  = 0;

  OCR1A = SR;            // set to run at above specified sampling rate
  TCCR1B |= (1 << WGM12);   // CTC mode
  TCCR1B |= (1 << CS12);    // 256 prescaler
  TIMSK1 |= (1 << OCIE1A);  // enable timer compare interrupt
  interrupts();             // enable all interrupts
}



void loop() {

// This sample program is interrupt driven, it automatically
// send the sampled Bio-Amp ADC converted signal to the serial port.

}

void readADC()
{

  data1= (analogRead(A0)*0.00488); // Sample Bio-Amp analog signal
}

// Timer 1 interrupt, automatically serviced every RATE ms.

ISR(TIMER1_COMPA_vect)
{
  Serial.print("E");
  readADC();
  Serial.print(data1);
  Serial.print ("\n");
}
```

Figure 7. Arduino sketch listing for the ECG Project.