

Heart Rate Monitor and Electrocardiograph Fundamentals

by: **Carlos Casillas**
RTAC Americas
Guadalajara
Mexico

1 Introduction

This application note demonstrates fundamental implementations of the cost effective heart rate monitor (HRM) and electrocardiograph (ECG or EKG) devices using Freescale products. The heart rate monitor is implemented on a low cost MC9RS08KA2 ultra low-end 8-bit microcontroller using an analog comparator module (ACMP) and standard IO pins. The electrocardiograph is implemented on the MC9S08JM60 USB device 8-bit microcontroller using an analog-to-digital converter (ADC) and USB connectivity to send data to a PC.

This document is intended to be used by biomedical engineers, medical equipment development engineers, or any person related with medicine and interested in understanding the operation of HRMs and ECGs. However, it is necessary to know the fundamentals of electronic analog and digital circuits.

2 Physiological Fundamentals

This section provides some fundamentals about heart physiology, helping to understand how the applications are developed.

2.1 Heart and Functions

The heart is the organ responsible for pumping blood throughout the body. It is located in the middle of the thorax, slightly offset to the left and surrounded by the lungs.

Contents

1	Introduction.....	1
2	Physiological Fundamentals	1
3	Heart Rate Monitor Implementation.....	5
4	Electrocardiograph Implementation.....	12
5	Running the HRM and ECG Demo.....	13
6	Optional Additions	19
7	Considerations and References.....	21
8	Conclusions.....	21

The heart is composed of four chambers; two atriums and two ventricles. The right atrium receives blood returning to the heart from the whole body. That blood passes through the right ventricle and is pumped to the lungs where it is oxygenated and goes back to the heart through the left atrium, then the blood passes through the left ventricle and is pumped again to be distributed to the entire body through the arteries.

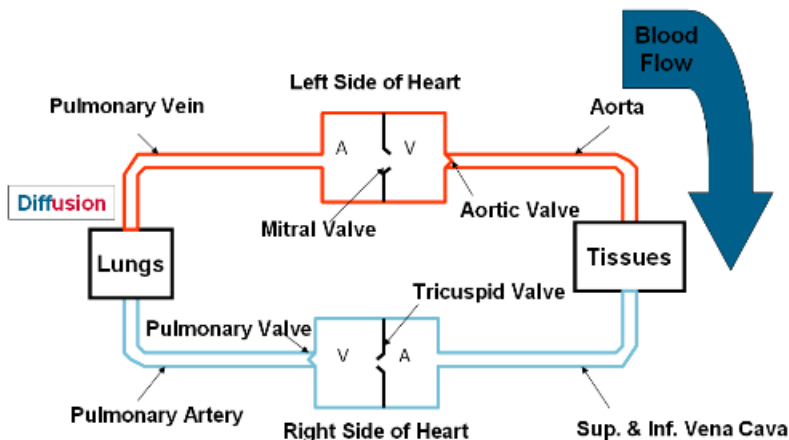


Figure 1. Blood circulation scheme

2.2 Heart—Electrical Activity

Electrical heart activity is based on depolarization and re-polarization of myocardial cells. The electrical impulse starts in the sinoatrial node (natural pacemaker) flowing through the atriums to reach the atrioventricular node and generating the atrium contraction. The current then flows through the His Bundle reaches the ventricles and flows through them generating the ventricular contractions. Finally, the current reaches the Purkinje fibers and re-polarization of the heart tissue occurs.

Figure 2 shows the heart chambers and tissues associated to electrical heart activity.

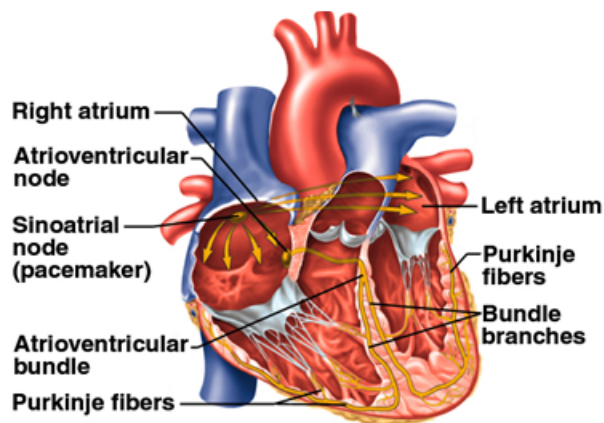


Figure 2. Cardiac conduction system

This is a list of events that occur in the heart on each heart beat. Figure 3 shows heart behavior and part of the generated signal also known as QRS complex:

1. Atrium begins to depolarize
2. Atrium depolarizes
3. Ventricles begin to depolarize at apex. Atrium repolarizes
4. Ventricles depolarize
5. Ventricles begin to repolarize at apex
6. Ventricles repolarize

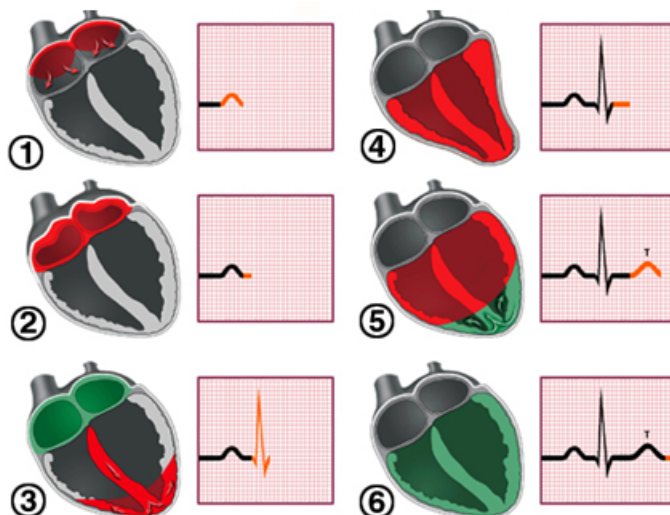


Figure 3. Myocardium electrical activity

2.3 Biological Electrical Potentials

The electrical potentials generated by the heart can be represented as vector quantity. For understanding purposes, the heart is represented as a dipole located in the thorax with a specific polarity at a certain moment, and an inverted polarity the next moment. The potential in a specific moment is defined by the amount of charge and the separation between charges.

Figure 4 shows an example of acquiring signals between two points on a muscle, using a differential operational amplifier.

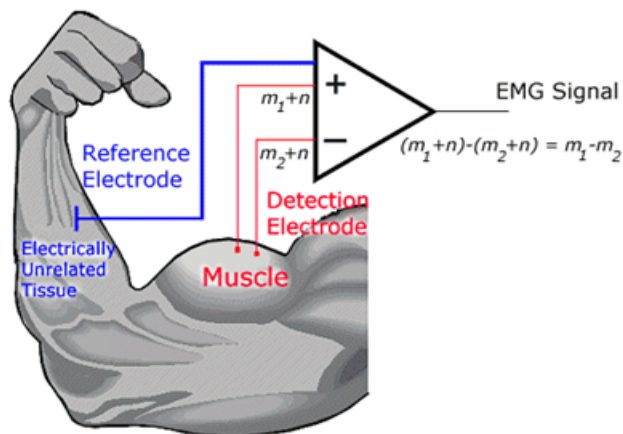


Figure 4. Differential amplification, m =signal from muscle, n =noise

As is shown in Figure 4, the output of the operational amplifier is the difference of magnitudes of m_1 and m_2 .

Each pair of electrodes or an electrodes combination is defined as lead. There are three basic leads used for cardiology. Lead I is at 0° , lead II is at 60° , and lead III is at 120° . The graphic representation of each lead is shown in Figure 5.

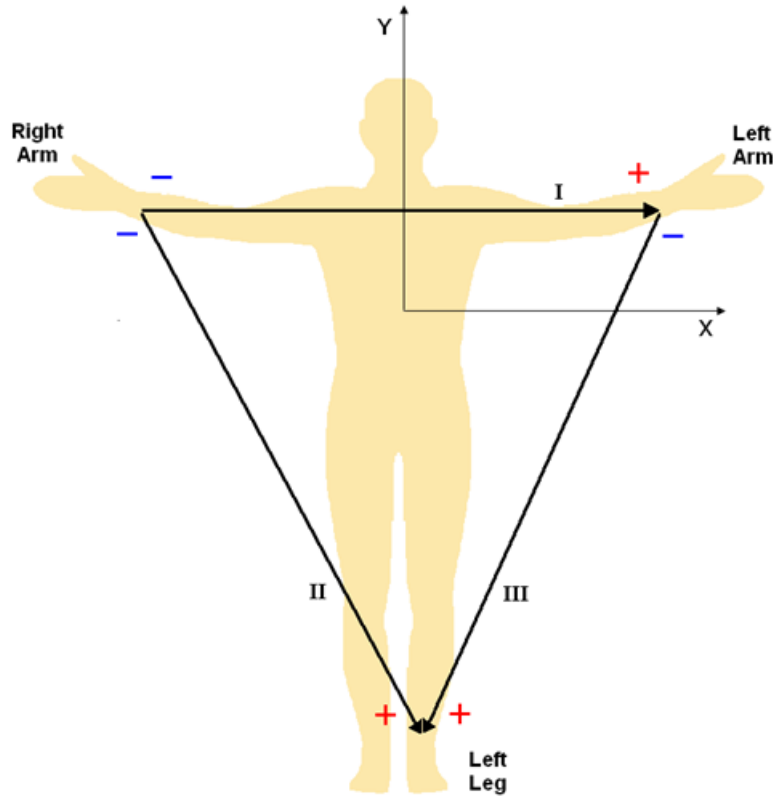


Figure 5. Einthoven Triangle

The three basic leads shown in [Figure 5](#) make-up the frontal-plane. Electrodes are placed on the limbs; left arm (LA), right arm (RA), and left leg (LL). Those connections are due to the legs and arms being used as a “wire” to detect the bio-potentials that occur in the chest.

2.4 Measuring the Heart Rate—Importance

Measuring the heart rate helps in different areas. For example; when used in fitness routines and indicating an exercise zone for different purposes such as fat burning. A normal heart rate depends on age. Children have higher heart rates compared to adults.

[Figure 6](#) shows a graph of the heart rate and the age for different purposes.

Figure 6. Exercise target zone chart



3 Heart Rate Monitor Implementation

The heart rate is the number of heart beats per minute. A heart rate monitor is a device used to measure the heart rate. This device helps to detect heart malfunctions such as a tachycardia, bradycardia, or other diseases.

Freescale solution includes the MC9RS08KA2 MCU. Key features used in this application are:

- ICS—Internal clock source avoids the use of the external oscillator and provides a bus clock of up to 10 MHz for fast code execution.
- MTIM—8-bit modulo timer is used to define time intervals used in heart beat detection algorithms.
- ACMP—Analog comparator has internal threshold levels and offers the possibility of adding an external reference.
- BDM—Background debug module provides a single-wire in-circuit debug interface.
- 6-pin and 8-pin packages—Small packages for small designs. Enough pins for heart rate monitor implementation.

Figure 7 shows the block diagram of an HRM implementation using the MC9RS09KA2 MCU.

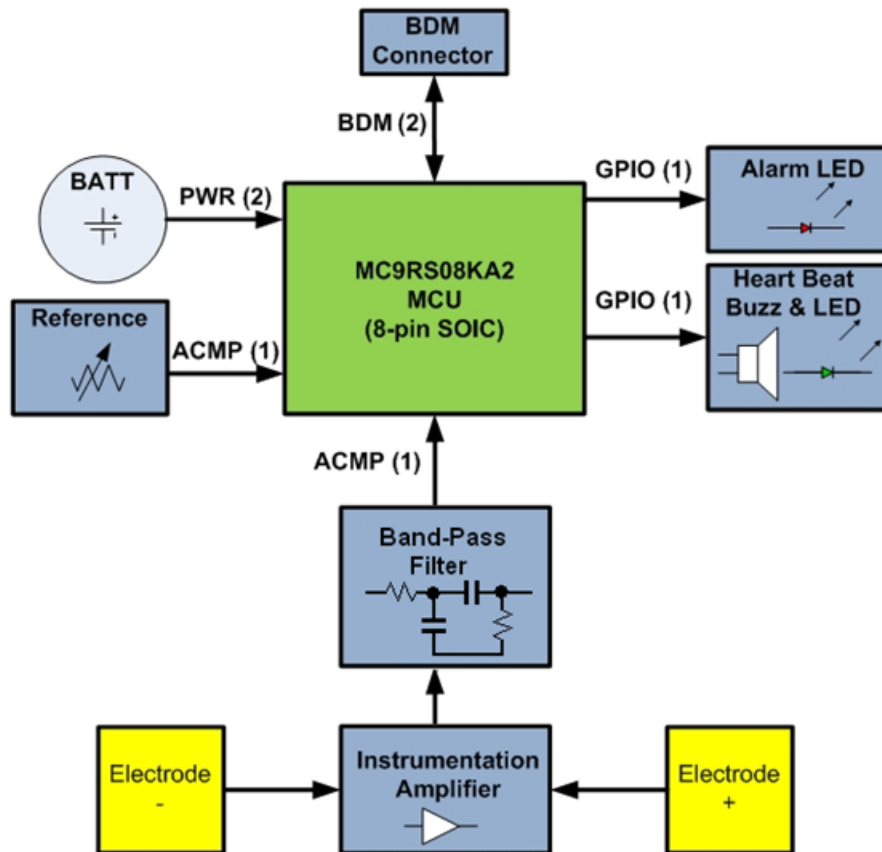


Figure 7. Heart rate monitor implementation using finger tips

3.1 Analog Front End

The signal acquisition is the first consideration when an HRM is implemented. The electrodes in the skin detect the small voltages in the order of hundreds of micro volts generated by heart activity, but the signal is too small and contains a lot of added noise.

Figure 8 shows a typical heart signal. In this signal, the heart muscles generate different voltages. The P wave represents the atrium contraction. QRS complex and the T wave represents the ventricles actions. Each time that this signal is present, a heart beat is generated. For this reason it is important to develop analog and digital signal conditioning. First, it is necessary to amplify the signal and filter the noise, and then extract the QRS complex.

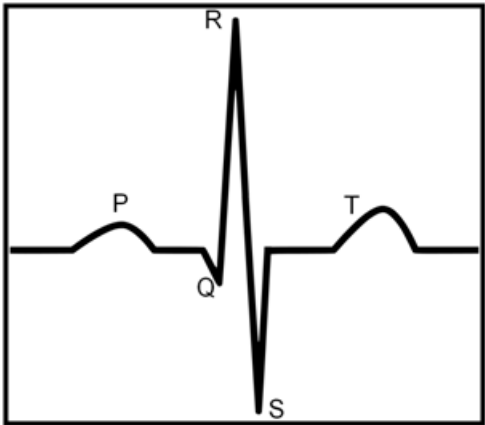


Figure 8. Typical heart signal

Noise and interference signals acquired in this type of system are caused by the electric installation. The small electrical signal from the heart generates a common-mode voltage and noise in the system. The signals from the heart are too small and it is necessary to amplify the signal and reduce the common-mode voltage on the system. Other aspects that generate noise are muscle contractions, respiration, electromagnetic interference, and electromagnetic emissions from electronic components.

Cardiac motion generates electrical signals with different potentials on the body that can be sensed with electrodes connected to the right hand and left hand and the reference electrode on the left leg or on the stomach.

The cut frequencies of the filter for this case are 0.5 Hz and 150 Hz. The lower limit of cut frequencies is due to noise that can be induced by respiration or by muscle contractions. The higher limit of the cut frequencies is defined to reduce the interference of the electromagnetic high-frequency signal. A normal heart operation is included in those frequency ranges.

The signal has an amplitude of approximately 1 mV peak to peak and contains common-mode voltage noises.

3.1.1 Amplification and Filtering

The electrodes of the left and right arms are connected to the inputs of an instrumentation amplifier; and the reference electrode of the left leg is connected to ground. **Figure 9** shows the mentioned connections using patch electrodes on the surface of the arms, and a reference electrode on the surface of one leg. The gain of the instrumentation amplifier (U32) is defined by resistors R86 and R49. In this case, a gain of 667.6 is implemented.

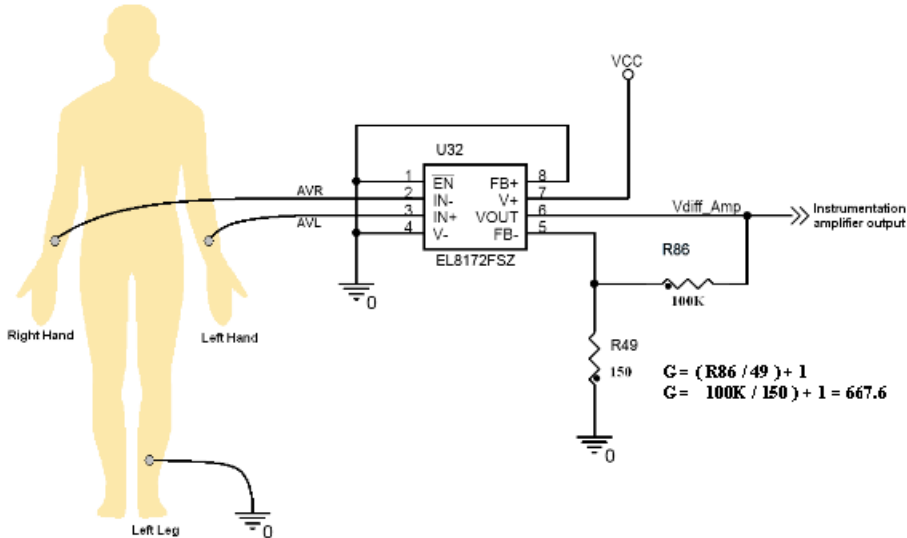


Figure 9. Electrode connections to the instrumentation amplifier

Heart Rate Monitor Implementation

The signal at the output of the instrumentation amplifier is filtered to reduce the noise. The equation to obtain the cut frequency of an RC filter is:

$$f_c = \frac{1}{2\pi RC}$$

For the cut frequency of 0.5 Hz, the defined values are the following:

$$R = 1\Omega$$

$$C = 330\text{nf}$$

$$f_c = \frac{1}{2\pi(330\text{nF})(1\text{M}\Omega)} = 0.482\text{Hz}$$

For the cut frequency of 150 Hz, the defined values are:

$$R = 47\Omega$$

$$C = 22\text{nF}$$

$$f_c = \frac{1}{2\pi(22\text{nF})(47\text{K}\Omega)} = 153.92\text{Hz}$$

The filters have a virtual ground applying an offset to the signal. This signal is sent to an operational amplifier that conditions the signal to better levels for the ADC microcontroller. The offset applied by the virtual ground helps to conserve the negative part of the signal, this is because the operational amplifier is a single supply device. See [Figure 10](#)

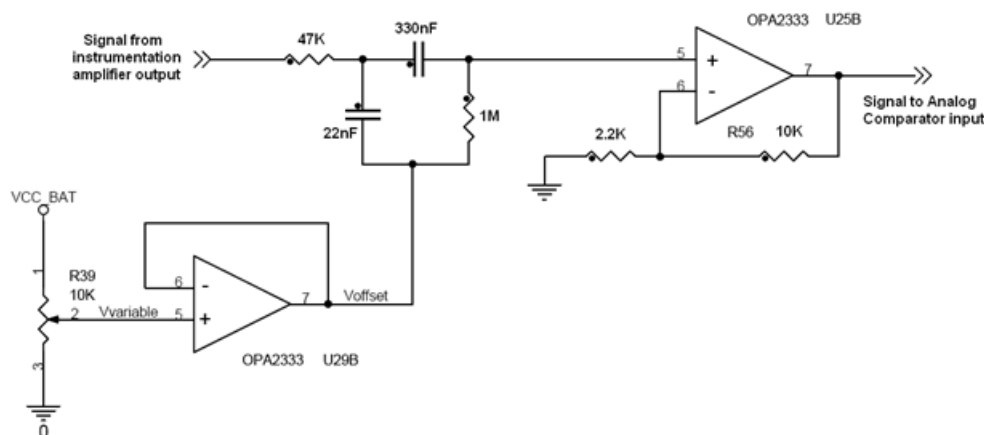


Figure 10. Filter and operational amplifier connections

The wanted ECG signal is periodical with an amplitude of about 3 to 4 Volts peak-to-peak.

NOTE

In this implementation, the offset voltage applied to the obtained signal is fixed and may at times need to be adjusted. However, this function can be implemented automatically through hardware and software implementations.

3.2 Software for the Heart Rate Monitor

This section describes the architecture flow chart and function descriptions for the HRM_KA project.

3.2.1 Architecture

The architecture of the HRM software is based on polling. This is because the MC9RS08KA family has no interrupt vector nor a stack pointer to provide interrupts. When the system starts, the general purpose pins are configured. One pin is used as output when a red LED is connected and is an alarm. Another pin is used as output where a green led and a speaker are connected to indicate visually and with sound when a heart beat is detected.

Another function initializes the ACMP to configure the use of an external reference and clear the flag used to indicate that the input signal passed the reference level detecting a heart beat.

The timer MTIM is then initialized to set its overflow flag each eight milliseconds. For polling this flag, a counter is incremented to count up to 1 second allowing to turn off the alarm and calculate the heart rate every second, turning on the alarm if the defined limit is passed. All the polling is inside an infinite loop.

3.2.2 HRM_KA Project Description

The HRM software includes all its functions in the main.c file, reducing the calls to subroutines. The flowchart of the main() function of the HRM_KA project is shown and the performed tasks by each part of the code detailed.

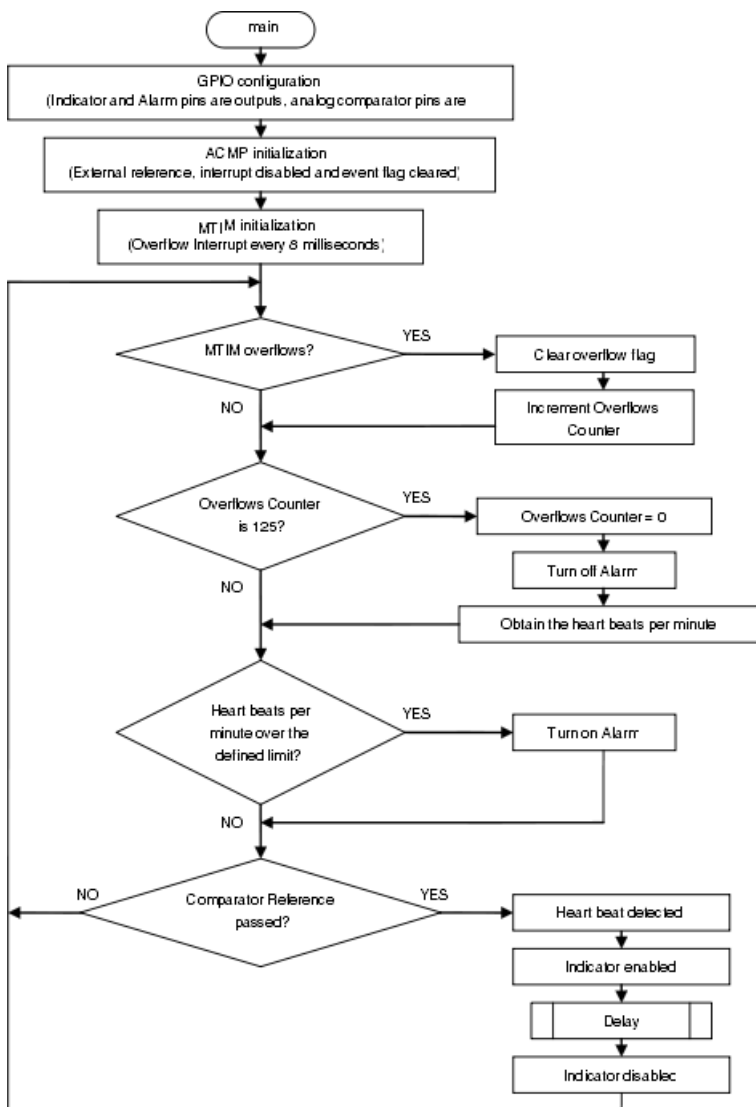


Figure 11. Flowchart of main() function of HRM_KA project

void main(void)

This is the principal function. It calls initialization functions to configure the GPIOs, MTIM, and ACMP. It then enters an infinite loop and polling status to execute sub-processes referred to as events.

```

void main(void)
{
  //byte lbFlag;

```

Heart Rate Monitor Implementation

```
//byte lbCounterPeak = 0x00;
//byte lb1SecCounter = 0x99;
unsigned int ul6Counter = 0, conttemp=0;

vfnGPIOConfig();
vfnACMPConfig();
vfnMTIMConfig();

for(;;)
{
    Endless loop. Each event is detailed separated.

    __RESET_WATCHDOG();          /* feeds the dog */
}
```

The initialization functions called by the main() are detailed below:

void vfnGPIOConfig (void)

This function configures the input and output pins required by the application. For the red and green leds, and the buzzer, the pins must be outputs. The external threshold reference and heart signal must be inputs. These configurations are in the function below:

```
void vfnGPIOConfig(void)
{
    PTADD_PTADD5 = 1;          //PTA5 as output INDICATOR
    PTAD_PTAD5 = 0;

    PTADD_PTADD4 = 1;          //PTA5 as output ALARM
    PTAD_PTAD4 = 1;

    PTADD_PTADD0 = 0;          //ACMP+ as input
    PTADD_PTADD1 = 0;          //ACMP- as input
}
```

void vfnACMPConfig (void)

This function configures the analog comparator to use an external reference for the threshold and clears the status flag.

```
void vfnACMPConfig(void)
{
    ACMPSC = 0x01;    //External reference -- Interrupt Disable -- Raising edge
    ACMPSC |= 0xA0;  //Enable the comparator and clear the event flag
}
```

void vfnMTIMConfig (void)

This function initializes the modulo timer. First, the counter is stopped and reset. Then the clock source is defined as the bus clock divided by 128. Finally, the timer is started.

The bus clock is 4 MHz, divided by 128 results in a frequency of 31.25 KHz, with a period of 32 microseconds. This is the time of one count of the timer, which is 8-bit. Therefore, its overflow value is 256. This is the reason why the overflow flag is set every 8 milliseconds. The initialization function is shown below:

```
void vfnMTIMConfig(void)
{
    MTIMSC = MTIMSC_TRST_MASK | MTIMSC_TSTP_MASK;
    // -- Reset Counter -- Counter stopped
    MTIMCLK = 0x07;    //Bus clock divided by 128
    MTIMSC_TSTP =0;    //The counter starts
}
```

Endless loop events

The following code explains the events that occur in the endless loop. These events are defined by the poll of flags and values of variables.

The first event checks if the overflow flag of the MTIM is set to clear and increment a variable named u16Counter. The code is shown below:

```
if (MTIMSC_TOF)          //Is set every 8ms
{
    (void)MTIMSC;
    MTIMSC_TOF = 0;
    u16Counter++;
    if (ACMPDelayEn)ACMPDelay++;
}
```

The next event checks if the u16Counter variable reaches the value of 125. This is because the variable is incremented 8 milliseconds each and multiplied by 125 the resultant time counted is one second. In this event, the u16Counter variable is cleared and the alarm is turned off. The average heart rate is calculated based on a buffer of 15 measurements. The code of this event is shown below:

```
if(u16Counter == 125)
{
    u16Counter = 0;
    ALARM = ALARM_OFF;
    u8Beat_SecondBuff[15] = 0;

    for(y=0;y<15;y++)
        u8Beat_SecondBuff[15] += u8Beat_SecondBuff[y];

    u8Beat_SecondBuff[15] = u8Beat_SecondBuff[15]<<2;
    //Multiply by 4 to get the total heartbeats in a minute

    u8Beat_SecondPtr++;
    if(u8Beat_SecondPtr==15) u8Beat_SecondPtr = 0;
    u8Beat_SecondBuff[u8Beat_SecondPtr] = 0;
}
```

The next event checks if the heart rate average is less the the defined limit (HRUpperLimit). If this is true, the alarm is turned on. This defines pediatric or adult limits, as is explained in Chapter 2.4 [Measuring the Heart Rate—Importance](#). The code is shown below:

```
if(u8Beat_SecondBuff[15]>=HRUpperLimit)    //Is HR over the limit?
{
    ALARM = ALARM_ON;
}
```

Next, the variable lbFlag takes the value returned by the subroutine bfnACMPRead() which is explained later. Then, the last event checks the value of the lbFlag variable. If this is true, the indicator LED is turned on and the buzzer beeps, then a delay is executed, and the LED and buzzer are turned off. The code is shown below:

```
lbFlag = bfnACMPRead();

if (lbFlag)
{
    INDICATOR = INDICATOR_ON;
    for(x=0;x<5000;x++)
    {
        __RESET_WATCHDOG();    // feeds the dog
    }
    INDICATOR = INDICATOR_OFF;
}
```

byte bfnACMPRead(void)

This function checks if the signal from the heart connected to the comparator passed the threshold. If true, it updates the buffer used to obtain the average of the heart rate and then increments a variable that counts the heartbeats. The code is shown below:

```
byte bfnACMPRead(void)
{
    if (ACMPSC_ACF)                //Passing the threshold ????
    {
```

Electrocardiograph Implementation

```

ACMPSC_ACF = 1; // Clear flag
if (ACMPDelay < 5) return (0);

ACMPDelay = 0;
ACMPDelayEn = 1;
u8Beat_SecondBuff[u8Beat_SecondPtr]++;
beats++;
return (1);
}
return (0);
}

```

Changing the maximum value of the heart rate to set the alarm

If you never reach the value of 100 heartbeats per minute to set the alarm change the maximum value for the alarm in the code is shown below:

```
#define HRUpperLimit 100
```

4 Electrocardiograph Implementation

Another important device used for cardiac diseases is the electrocardiograph. This device plots a graph of the electrical activity of the heart. It is useful in diagnostic and therapeutic areas helping detect abnormal heart functions such as arrhythmia, tachycardia, bradycardia, or other cardiac problems.

4.1 Electrocardiograph Software

The software required for this implementation consists in the USB data logger based on the MC9S08JM60, which is explained entirely in the application note titled *The USB Data Logger Based on the MC9S08JM60* (document AN3582). This application note also includes a graphical user interface (GUI), with details in Chapter 5.5 [Using the Freescale JM60 GUI to Obtain the ECG Signal](#).

4.2 Hardware Connection

The analog front end used by the HRM is used by the electrocardiograph. Not only is it compared with a reference level as implemented in the HRM, but the signal is sampled by the analog-to-digital converter to capture the wave form allowing to send through a USB communication.

The Freescale solution includes the MC9S08JM60 MCU. Key features for this application note are the following:

- USB—USB 2.0 full-speed (12 Mbps) device controller with a dedicated on-chip USB transceiver.
- SCI, SPI, and I²C modules—Two serial communication interfaces, two serial peripheral interfaces, and inter-integrated circuit bus allowing to expand communication and connectivity options.
- ADC—Analog-to-digital converter 12-channel and 12-bit with automatic compare function and an internal temperature sensor.
- BDM—Background debug module provides a single-wire in-circuit debug interface.

The following block diagram shows the specific blocks implemented in this application. The dashed line indicates an optional block that can be included.

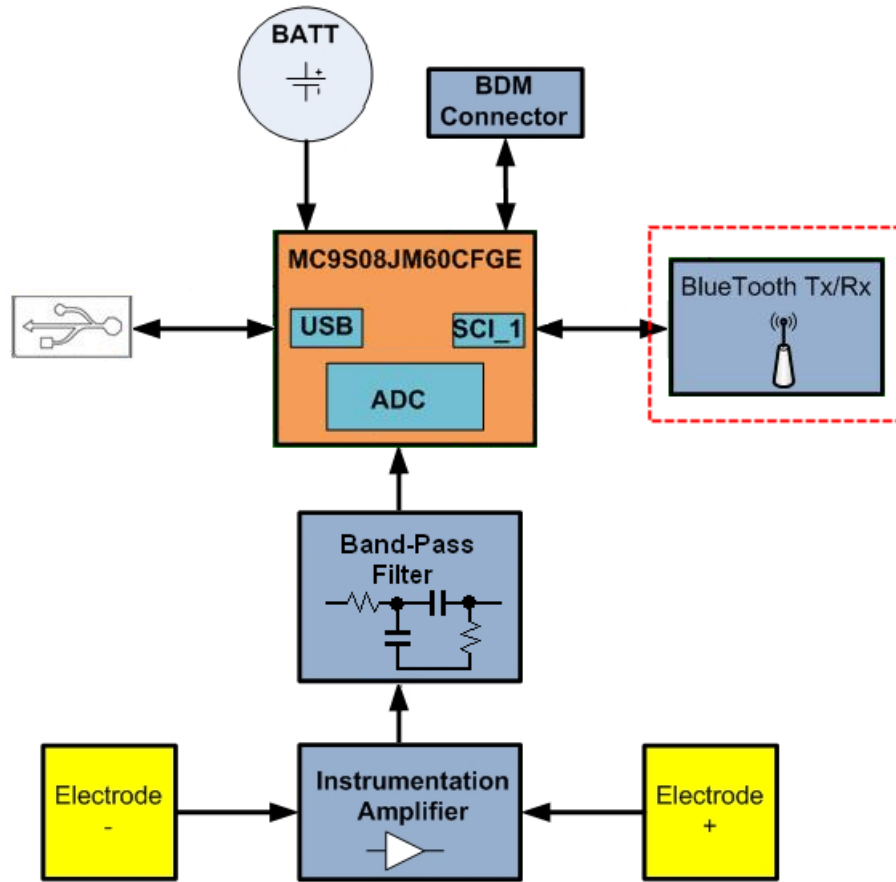


Figure 12. MC9S08JM60 1-lead electrocardiograph implementation

5 Running the HRM and ECG Demo

This section provides considerations, connections, test results, and tool usage for the HRM and ECG demo.

5.1 Settings and Considerations

To obtain the heart rate using the Freescale solution, steps are required. These steps are listed below:

1. Connect the board to a USB port on a host computer. If the JM60 is programmed, the PC requires a driver. This is included with the installation of the Freescale JM60 GUI included in the zip file of this application note.
2. Make sure that switch SW1 is in the down position to power the system using a USB cable. [Figure 13](#) shows the position of this switch.
3. The KA2 microcontroller must be programmed with the CodeWarrior project HRM_KA included in the AN4059SW.zip file that can be downloaded from the Freescale web page. Connect a BDM to a USB port on the computer, and connect it to the closest header to the KA2.
4. The JM60 microcontroller must be programmed with the CodeWarrior project datalogger included in the AN4059SW.zip file that can be downloaded from the Freescale website. Connect a BDM to a USB port on the computer, and connect it to the closest header to the JM60.

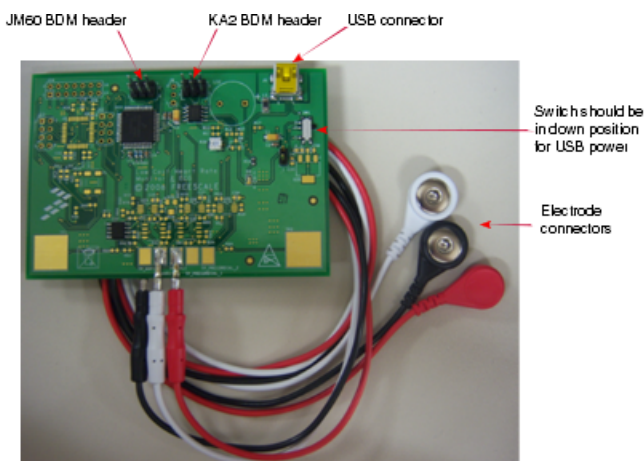


Figure 13. Freescale solution considerations

5.2 Connecting Electrodes

As is explained in Chapter 2.3 [Biological Electrical Potentials](#), the electrodes can be collocated in different places of the body, depending on the signal that it is looking for. [Figure 14](#) shows the placement of the electrodes on the body:

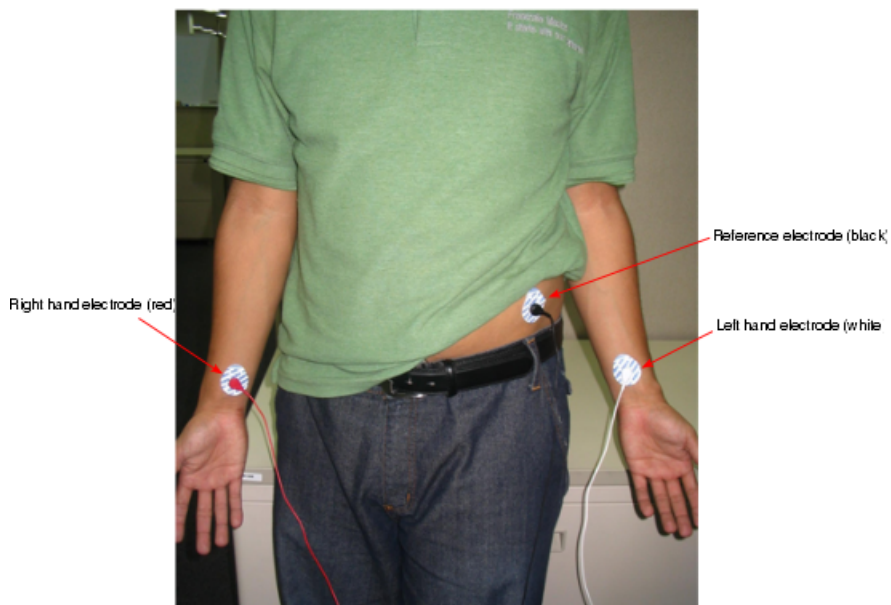


Figure 14. Electrodes placement implemented

5.3 Obtaining Heart Rate by Using a Visualization Tool

The CodeWarrior project includes a lot of tools to help debugging projects. The project for the KA2 allows visualizing the average of heart beats obtained in one part of the program. To enable this function go to the debugging window shown in [Figure 15](#).

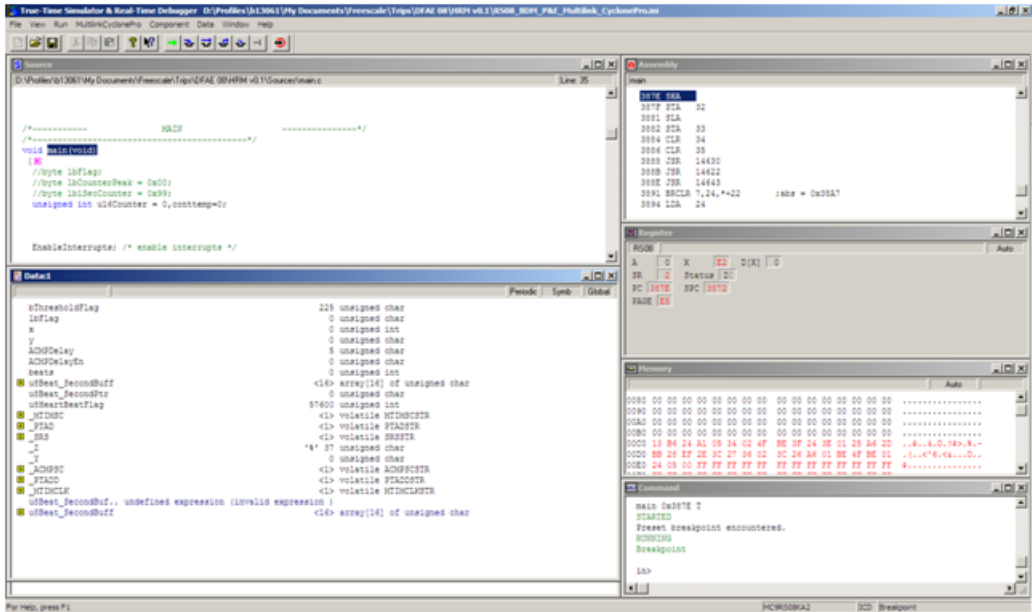


Figure 15. Debugging window

Then, go to the **Component** menu and click on **Open**. The following window opens [Figure 16](#):

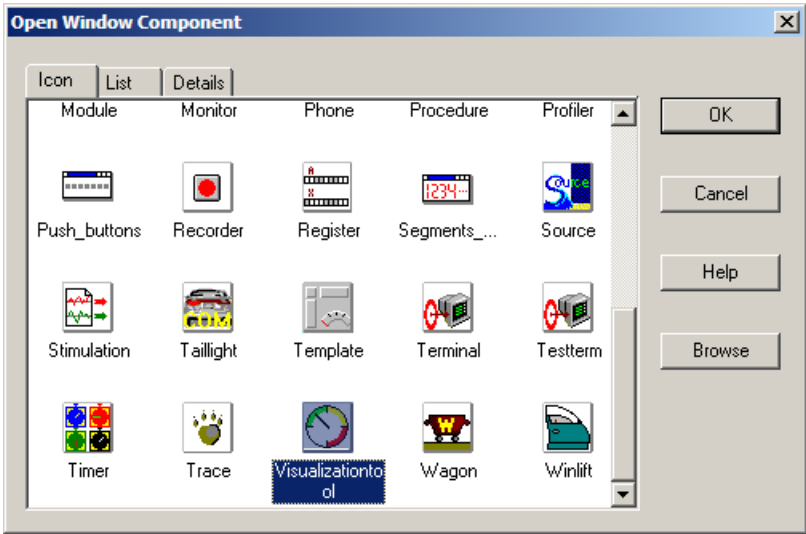


Figure 16. Component window

Click on the **OK** button to close the window. Then, click on the open folder button [Figure 17](#):



Figure 17. Open folder

In the window that appears in the **Open Visualization Tool Layout**, look for the file HRM.vtl. This file is in the bin folder, the folder that includes the entire project. Select this file as is shown in [Figure 18](#), and click on **Open** button.

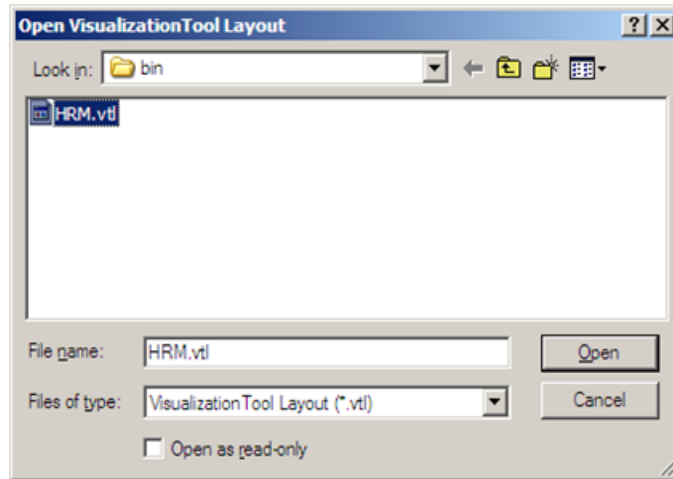


Figure 18. HRM

Finally, run the code and look at the heart rate shown on the screen.

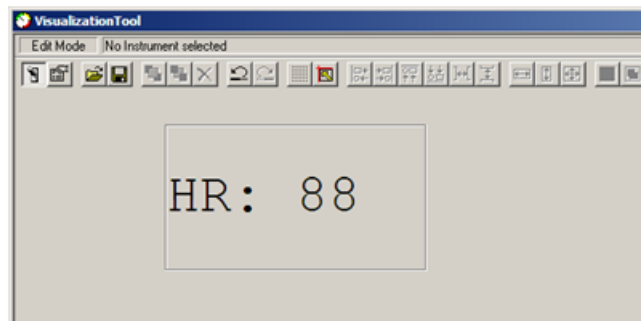


Figure 19. Heart rate code

5.4 Measuring the ECG Signal with the Oscilloscope

The following figures show the waveforms of the obtained signal in each part of the analog front end. [Figure 20](#) shows the waveform at output of the instrumentation amplifier.

The peak-to-peak voltage of the heart signal is in the order of about 650 mV, and with noise added.

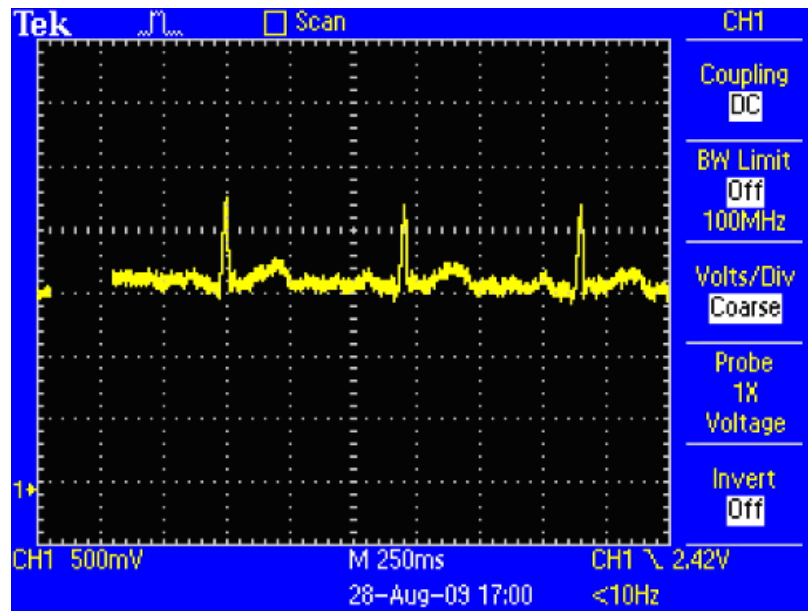


Figure 20. Heart signal at the output of the instrumentation amplifier

Figure 21 shows the heart signal at the output of the passive band-pass filter.

The noise present in the signal has been reduced; however, the peak-to-peak amplitude has been also attenuated due to filter effects. To counteract this effect and condition the signal to values of rail-to-rail voltages of the ADC, a second amplification is applied using an operational amplifier.

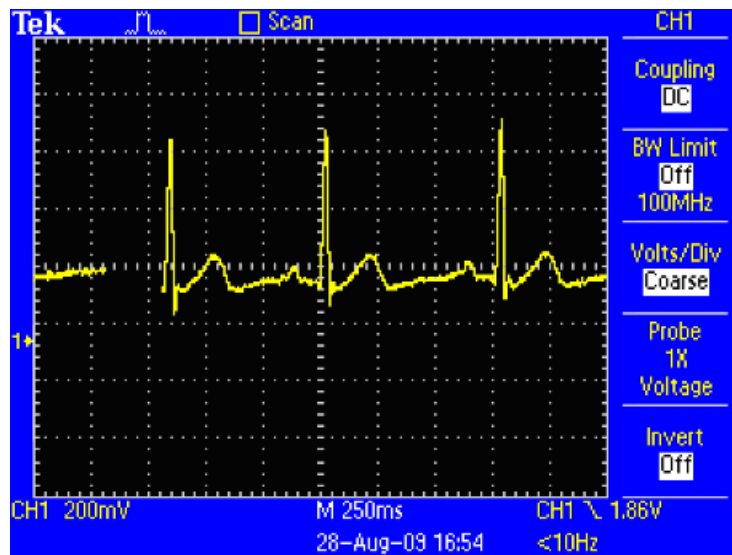


Figure 21. Heart signal at the output of the band-pass filter

The signal at the output of the operational amplifier is shown in Figure 22. The signal has been amplified, and its peak-to-peak voltage is in the order of 3 V. This is an optimal value to be passed to the ADC

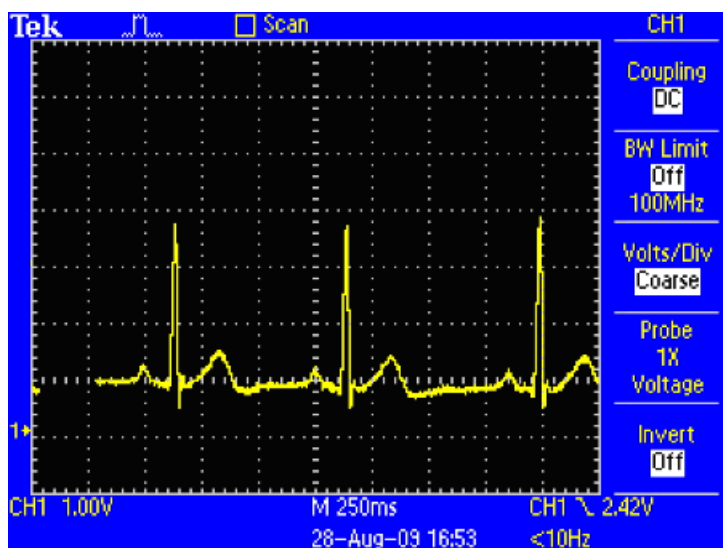


Figure 22. Heart signal at the output of the secondary operational amplifier

5.5 Using the Freescale JM60 GUI to Obtain the ECG Signal

Open the JM60 GUI. The main window is shown below:

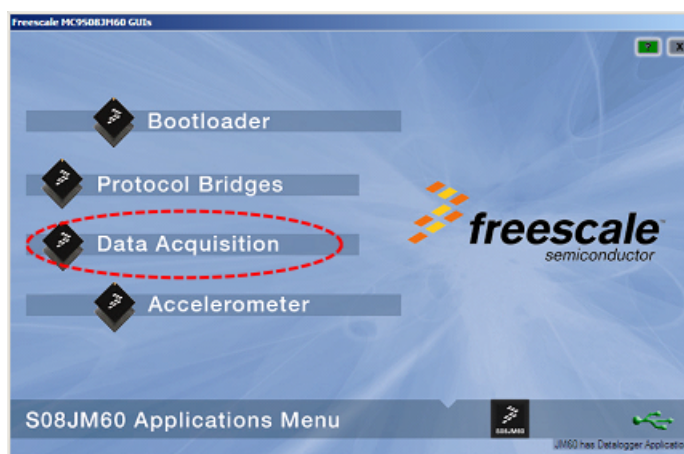


Figure 23. S08JM60 Applications menu

Select the Data Acquisition option and the following window opens.

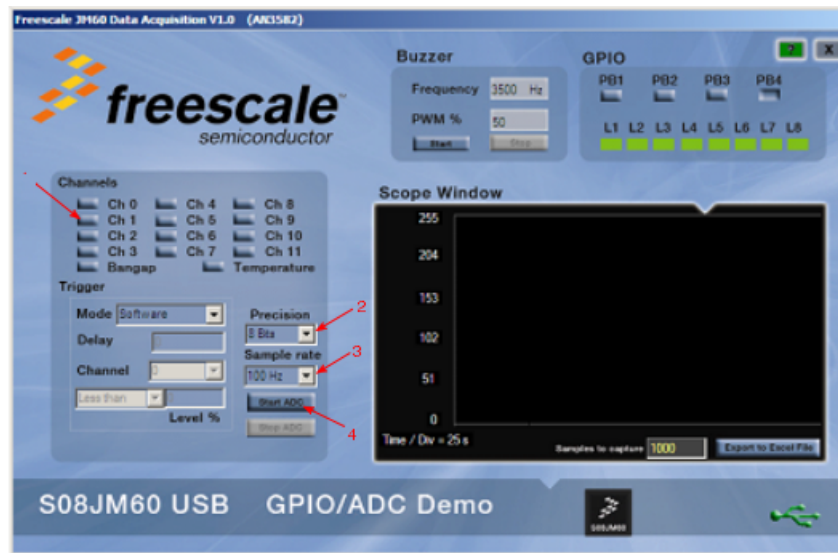


Figure 24. Data acquisition window

Select :

1. Ch1
2. Set the precision to 12 Bits
3. Set the sample rate to 200 KHz
4. Then, press “Start ADC” button

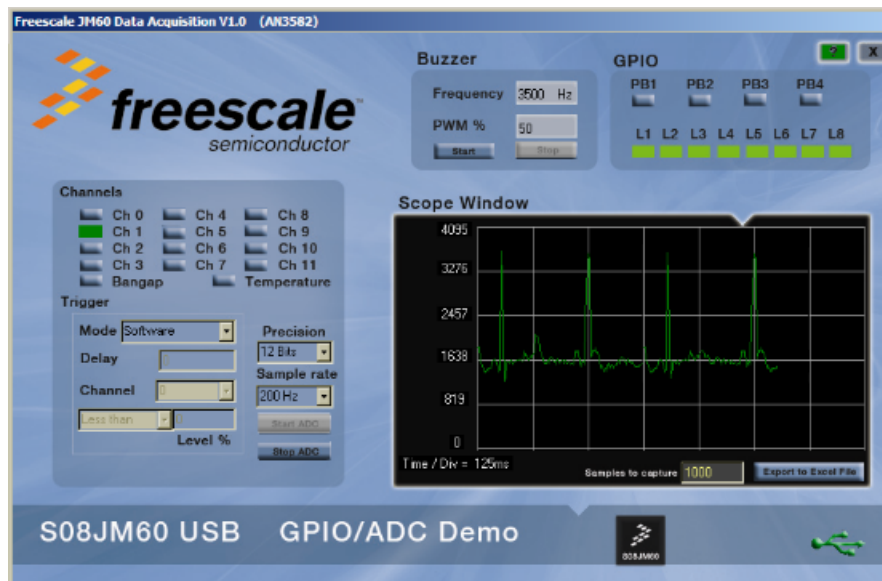


Figure 25. S08JM60 USB GPIO/ADC demo

The heart signal is shown in the **Scope Window**. For complete use of the JM60 GUI, refer to the application note titled *The USB Data Logger Based on the MC9S08JM60* (document AN3582). It can be found on the [Freescale webpage](#).

6 Optional Additions

This section shows additions that can be implemented to the demo.

6.1 USB Personal Healthcare Device Class Compatibility

A USB Personal Health Care Device Class specification (PHDC) was required to implement proprietary methods for sending personal healthcare data to a USB host. The USB PHDC is to enable seamless interoperability between personal healthcare devices and USB hosts that allow for a variety of new cases for both devices and hosts.

Freescale’s Medical Applications USB stack is based on the USB’s PHDC and IEEE-11073. It is compatible with the Continua Host emulator software available on the Continua Health Alliance webpage. This Medical USB stack enables microcontrollers with guidelines for the Continua Health Alliance connectivity and is the first step to test your application prior to the official Continua Certification. The Continua Host Emulator can be downloaded for free for Continua Health Alliance members.

Freescale provides CodeWarrior projects to implement a PHDC using the MC9S08JS, MC9S08JM 8-bit devices, and the MCF51JM 32-bit Coldfire V1 device. For more information, please refer to the [Freescale Medical webpage](#).

6.2 Bluetooth Connectivity

Bluetooth is an open wireless protocol for exchanging data over short distances from fixed and mobile devices creating personal area networks (PANs). It was originally conceived as a wireless alternative to RS232 data cables. It can connect several devices overcoming problems of synchronization. Bluetooth provides up to 10 meters of distance to set-up communication with a speed of up to 1 Mb/s. Bluetooth also provides high compatibility to most computers and it is not necessary to implement a special network for communication.

This kind of communication can be implemented on the HRM and ECG to connect wirelessly. The National Instruments LMX9838 device is optimal for this application.

The hardware configuration can be implemented as is shown in [Figure 26](#).

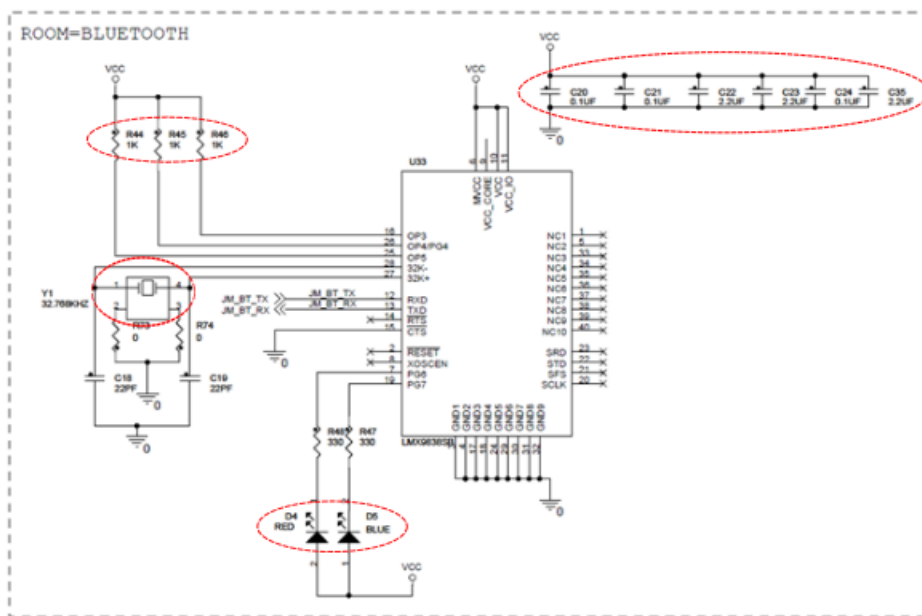


Figure 26. Hardware connection for Bluetooth implementation

6.3 Baud Rate Configuration

The LMX9838 provides a UART interface to communicate with the MCU through the SCI module LMX9838 that must be configured with a baudrate. The UART interface supports formats of 8-bit data with or without parity and with one or two stop bits. It can operate at standard baud rates from 2400 bits/s up to a maximum baud rate of 921.6 kbits/s. The UART baudrate is configured during startup by checking option pins OP3, OP4, and OP5 as shown. Table 1 shows different UART frequency settings.

Table 1. UART frequency settings

OP3	OP4	OP5	Baud Rate
1	0	0	Read from NVS ¹
1	0	1	9.6 kbps
1	1	0	115.2 kbps
1	1	1	921.6 kbps

When the value is 1 this indicates that a 1 K Ω pull up resistor must be placed.

An implementation example of this communication option can be found in the application note titled *Implementing Glucometer and Blood Pressure Monitor Medical Devices* (document AN4025). That application note shows the connection of the Bluetooth transceiver to an MC9S08LL16, and provides the code used for communication.

7 Considerations and References

- The use of these kinds of devices do not supply the professional diagnostic of a physician. Implementing this device helps to detect if a diseases is occurring, but the treatments must be administrated by the cardiologist.
- For more information about Freescale microcontrollers, refer to the *MC9RS08KA2/1 Datasheet*, and the *MC9S08JM60 Datasheet*, that can be found on the [Freescale webpage](#).
- The software was developed and tested with CodeWarrior v6.2.1
- Download the tools and source files for AN4059SW.zip from www.freescale.com.
- A variety of videos showing the implementation of explained medical devices can be found on www.freescale.com/medical and www.youtube.com/freescale.

8 Conclusions

Freescale products catalog offers many devices to implement medical applications. The MC9RS08KA2 and MC9S08JM60 are examples of this by offering high performance, low cost and low power consumption in a variety of ultra-low-end microcontrollers, and USB device microcontrollers.

This system is a low-cost, low-power, and battery operated with Continua Health Alliance connectivity over USB. It is uses only a 1-lead. However, a full 12-lead ECG can be implemented populating all the hardware, and using the digital signal controller (DSC) MC56F8013 to execute algorithms to detect arrhythmias or other heart diseases.

How to Reach Us:

Home Page:

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see <http://www.freescale.com> or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to <http://www.freescale.com/epp>.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc.2010. All rights reserved.