Wireless Musical Electrocardiogram

Amy Tang and Sinit Vitavasiri
Motivation

• Wireless communication for medical applications is able to solve clinical needs and risks, while providing the patient with the freedom of movement.
Objectives

• Obtain EKG from patient
• Intelligently analyze, store, and transfer data to end user
• Ability to detect varying conditions of patient.
Modes of Operation

Mode 1:
• EKG Wireless Monitor
• Bright LED “Beat” Indicator
• Digital Heart Rate Display

Mode 2:
• Detecting Abnormality from Heart Rate
• Detecting Abnormality from EKG

Mode 3:
• Heart-Rate-Controlled Music
EKG → Gain → A/D → Transmitter → Receiver

Synchronizer → Noise Filtering → BPF1 → BPF2

Heart Beat Detector

# posedge in 10 secs × 6

Abnormality Detector

Tempo Controller

D/A → oscillo scope

Music Data

LED Display

LEDs

Audio

Reset Clock SW

heart_rate
Typical EKG Waveform

- **P**: Atrial Depolarization
- **QRS**: Ventricular Depolarization
- **T**: Ventricular Repolarization
Wireless Transceiver

CC1010 (ChipCon)
- 8051 Compatible Microcontroller
- 300-1000 MHz RF Transmission
- 32 kB Flash Memory
- A/D Converter (10 bit)
Wireless – Calibration Algorithm

Start single calibration

Write FREQ_A, FREQ_B

Write FRMAIN:
RXTX = 0; F_REG = 0; RX_PD = 0;
FS_BD = 0

Write CURRENT.VCO_CURRENT = RX current
Write PLL.REFDIV = RX reference divider

Write RFMAIN:
RXTX = 1; F_REG = 1
RX_PD = 1; TX_PD = 0; FS_PD = 0
RESET_N = 1

Write CAL.CAL_START = 0

Write CURRENT.VCO_CURRENT = TX current
Write PLL.REFDIV = TX reference divider

Write CAL.CAL_START = 1

Write FRMAIN:
RXTX = 0; F_REG = 0; RX_PD = 0;
FS_BD = 0

Wait for 38 ms or Read CAL or wait until CAL.CAL_COMPLETE = 1

Write CAL.CAL_START = 1

End of Calibration

Frequency register A is used for RX mode, register B for TX

Frequency register A is calibrated first

‘RX current’ is the VCO current to be used in RX mode

Calibration is performed in RX mode, Result is stored in TEST0 and TEST2, RX register

‘RX current’ is the VCO current to be used in RX mode
Mode 1: Noise Filtering

Noisy signal from wireless receiver

3 Hz 22.5 Hz

3 Hz 10 Hz

Attenuate peaks, preserve signal features

Smooth out peaks

Filtered data

Use Matlab to determine the impulse response for each filter
Mode 1: Noise Filtering

- **Divider**
  - sample
- **Synchronizer**
  - reset
- **Major FSM**
  - AU_start
  - AU_busy
- **AU**
  - multiplier
  - DAC_start
  - analog_busy
  - int_data
- **SRAM**
  - read_data
- **D/A (AD 558)**
  - ext_data
- **Received Signal**
- **ROM1**
  - rom_addr
  - rom_q
- **ROM2**
  - sram_we
  - sram_addr
  - sram_q
- **Control Signals**
  - clk (global)
  - reset
  - reset sync (global)
  - cs_bar_DA
- **Additional Components**
  - 8 int_data
  - 12 rom_addr
  - 16 rom_q
  - 6 sram_addr
  - 8 sram_addr
  - 8 sram_q

**Mode 1: Noise Filtering**
Mode 1: Heart Beat Detector
Mode 1: Heart Rate

- Count number of positive edges in 10 seconds
- Heart rate = # count × 6
Mode 2: Detecting Abnormality

- Normal heart rate range = [50, 200]
- If heart rate is out of this range, the LED will illuminate.

- Normally, $T > P$
- Compare the two peaks from stored data in SRAM
- If $T < P$, the LED will illuminate.
Mode 3: Music from the Heart

- Music tempo is controlled by the heart rate
- Change tempo without changing pitch

<table>
<thead>
<tr>
<th>Location 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location 2</td>
</tr>
<tr>
<td>Location 3</td>
</tr>
<tr>
<td>Location 4</td>
</tr>
<tr>
<td>Location 5</td>
</tr>
<tr>
<td>Location 6</td>
</tr>
<tr>
<td>Location 7</td>
</tr>
<tr>
<td>Location 8</td>
</tr>
<tr>
<td>Location 9</td>
</tr>
<tr>
<td>Location 10</td>
</tr>
</tbody>
</table>
Mode 3: Algorithm

function output = timescale(sig, compression, maxfreq)
% takes in a signal in the time domain and scales its length, thus increasing its tempo.
% It scales the signal by compression, where compression is less than 1.
% It takes in maxfreq in order to compute how often to remove samples.

if nargin < 3, maxfreq = 4096; end
n = length(sig);
% Computes how often to remove samples
Timediv = floor(.08*maxfreq*2)
% Computes how many samples to remove
remove = floor((1-compression)*timediv)
output = 0;
% Remove samples, and recombine signals
for i = remove+1:(timediv±remove):(n-timediv)
    output = [output; sig((i-remove):(i+timediv-remove))];
end
Mode 3: Music of the Heart

Heart Rate Range
- 51-80
- 81-110
- 111-140
- 141-170
- 171-200