|  |  |
| --- | --- |
| **ITU – Telecommunications Standardization Sector**STUDY GROUP 21 Question C/16 (ex-Q6/16)**Video Coding Experts Group (VCEG)**75th Meeting: 2-8 November 2024, Kemer, TR | Document VCEG-BW13-v1 |

|  |  |
| --- | --- |
| Question: | C/16 SG21 (VCEG)  |
| Source: | **Viktor Herrmann, Philipp Kreowsky, Benno Stabernack, Jonathan Pfaff, Heiner Kirchhoffer, Christian Helmrich, Sophie Pientka, Christian Rudat, Heiko Schwarz, Detlev Marpe, Thomas Wiegand (Fraunhofer HHI)** |  Email: | firstname.lastname@hhi.fraunhofer.de  |
| Title: | **Demonstrator for Fraunhofer HHI’s response to the Call for Proposals on the compression of biomedical waveform data** |
| Purpose: | Information |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Introduction

This contribution presents a demonstrator adapting the methods of Fraunhofer HHI’s response to the Call for Proposals on the Coding of biomedical waveform data issued by VCEG. The presented system aims to show the computational requirements of the approach in relation to modern embedded devices such as wearables and smart watches.

For this purpose, the encoder is deployed on an exemplary single-board computer device - the Raspberry Pi Zero 2W - utilizing a 64-bit quad-core processor clocked at 1GHz with similar performance to the aforementioned devices.

It is reported that encoding methods can be successfully implemented on low-end computational devices utilizing around 5%-16% percent of a single core CPU resources at a memory footprint of up to 5.8 Mbyte

# Description of the Demonstrator Setup

## Overview

The schematic setup of the demo is shown in Figure 1. On the client side (encoding device), raw waveform is sampled using a heart rate sensor and locally displayed. This data is encoded with the codec proposed in Fraunhofer HHI’s response to the Call for Proposals on the coding of biomedical waveform data. The compressed data is sent to the host system via network connection. The host system decodes the compressed data and displays it on the host screen.



*Figure 1: Toplevel setup of the demonstrator with encoder, pulse sensor and display on embedded devices.*

## Device

The Raspberry Pi Zero 2W is a single-board computer with flexible I/O breakout pins and various communication interfaces such as 2.4GHz IEEE 802.11b/g/n wireless LAN. It uses a Broadcom BCM2710A1 ARM cortex-A53 64-bit quad-core processor, clocked at 1 GHz and is equipped with 512 MB of LPDDR2 SDRAM. The Raspberry Pi runs the 64-bit version of Debian GNU/Linux 12 (bookworm) operation system (OS).

In Table 1, the raw computational resources are compared to modern smart watch devices. The main SoC of Raspberry Pi Zero 2W uses a comparable architecture with similar clock rate and number of cores. The memory is scaled at around 0.5-0.25 %of the compared devices.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Device | Release  | SoC | CPU | RAM[MB] | Technology | Source |
| Raspberry Pi Zero 2W | October 2021 | Broadcom BCM2710A1 | Quad-core @ 1GHz ARM Cortex®-A53 | 512 | - | [2] |
| Samsung Galaxy Watch3 | August 2020 | Exynos 9110 | 1.1GHz Dual-core ARM Cortex®-A53 | 1024 | 10nm | [3] |
| Samsung Galaxy Watch7 | July 2024 | Exynos W1000 | Cortex®-A78 1.6GHz Single-core + Cortex®-A55 1.5GHz Quad-core | 2048 | 3nm GAA | [4] |

*Table 1: Comparison between Raspberry Pi Zero 2W and modern smart watches*

## Encoding Setup and I/O on Embedded Client

Single channel input data is generated by the Optical Pulse Sensor (Iduino SE050) which is connected to a 16-bit TI ADS1115 ADC. The ADC can be operated with several available sampling rates (8, 16, 32, 64, 128, 250, 475, 860 Hz). The original waveform can be shown on a local display. For the experiments, sampling rates of 250, 475, and 860 Hz are used. Besides sampled data, the encoder can also be run using offline data from datasets, where the latter can have multiple channels.

The client application invokes the encoder which is compiled from C++ source using gcc-12.2.0 with optimization level -O3 as a single-threaded subprocess that runs on a separate processor core to isolate the processing from the sampling and display process.

The application uses the standard in- and output stream interface to provide the raw heart rate data to the encoder and read the compressed bitstream. The compressed bitstream is divided into TCP-packages which are sent to the host application.

## Decoder and Host System

The host system deploys the decoder described in VCEG-BW02 and is configured as an access point so the client can reach the server via 2.4GHz IEEE 802.11b/g/n wireless LAN. The host system application runs the server side of the TCP connection that receives and handles the TCP packages from the client application. The data rate of the encoded bitstream is measured and averaged over several consecutive packages. The decoder is also isolated as a subprocess that uses standard in- and output stream interface for data communication. The bitstream is then decoded and the reconstructed signal displays on a local display.

## Encoder Configuration

The encoder executable of this demo is equal to the encoder executable used for the generation of the results for VCEG-BW02. However, a config file was used that yields a lower encoder runtime with a slightly deteriorated compression performance compared to the encoder configured to generate the bitstreams for VCEG-BW02. A comparison of the two configurations is given in Fig. 2. The following modifications were made:

* For the block matching mode, the multi hypothesis prediction only tests combinations of the best few single hypothesis candidates. Fewer filtering modes than in the default preset are tested. The pre-search results in a maximum of two single hypotheses and two multi hypothesis candidates for which the full transform coding is tested.
* The sample-wise linear predictive filtering is limited to filter order 4.
* The block sizes are limited to 256 and 128 without block merging.

|  |  |
| --- | --- |
|  |  |

*Figure 2: Comparison of coding efficiency for the MIT ECG dataset with default- and fast- encoder configurations. The plot shows exemplary recordings (left: 100\_x16, right: 118\_x16). The encoding time corresponding to each curve is plotted as a dashed line in the same color.*

# Experimental Results

## Throughput with Maximum Utilization with Offline Dataset

To investigate the maximum throughput, the encoder on the client side uses offline data from a standard dataset that is immediately available for processing. With this setup, it is aimed to evaluate the maximum performance of the encoder on a single thread. The encoder runs without the sampling application. For the throughput measurement, the MIT ECG dataset with 2-channel 16-bit data at 300 Hz sampling rate is used. The results in Table 2 show a throughput of 3.8-5.5 k samples per second depending on the selected step size for the QP.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Dataset | Samples  | Step size for QP | Runtime | Samples/s |
| MIT ECG 100\_x16.dat | 1300000 | 1.001 | 5m 22.4s | 4032.25 |
| MIT ECG 118\_x16.dat | 1300000 | 1.001 | 5m 35.2s | 3878.28 |
| MIT ECG 100\_x16.dat | 1300000 | 16 | 4m 1.8s | 5376.34 |
| MIT ECG 118\_x16.dat | 1300000 | 16 | 4m 4.9s | 5308.28 |
| MIT ECG 100\_x16.dat | 1300000 | 128 | 3m 53.0s | 5579.40 |
| MIT ECG 118\_x16.dat | 1300000 | 128 | 3m 55.6s | 5517.82 |

*Table 2: Throughput measurement for offline data.*

## Utilization with Online Data

Following the results of 3.1, it can be assumed that the encoder is real-time capable. The full demonstrator is run using the aforementioned fast configuration. Table 3 shows the measured computational resource utilization. Depending on the rate, the encoder runs at an average utilization of 4.5, 8.4 and 15.8 percent for the sample rates 250, 465 and 860 Hz respectively.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rate [Hz] | Avg. CPU [%] | Max CPU [%] | Avg. RAM [MiB] | Peak RAM [MiB] |
| 250 | 4.5  | 14.0 | 4.4 | 4.9 |
| 475 | 8.4 | 22.0 | 4.8 | 5.8 |
| 860 | 15.82 | 42.0 | 4.9 | 5.8 |

*Table 3: CPU and RAM utilization of the encoder process on the Raspberry Pi Zero 2W.*

# Summary

This contribution reports a real-time demonstration for the codec system described in Fraunhofer HHI’s response VCEG-BW02 to the Call for Proposals on the coding of biomedical waveform data. An encoder configuration is used which yields a compression performance that is only slightly worse than the performance achieved with the encoder configuration used for BW02. With this configuration, the encoder is deployed on an exemplary single-board computer device with a performance that is similar to modern wearable devices such as smartwatches. The results show that encoding on this low-end device is real-time capable for up to 3.8-5.5k samples per second. With the presented setup using real-time sampling at 250, 475 and 860 Hz, an average CPU utilization of 4.5, 8.4 and 15.8 percent is reported.

# References

1. Raspberry Pi Foundation, "Processors," *Raspberry Pi Documentation*, Accessed: Oct. 30, 2024. [Online]. Available: <https://www.raspberrypi.com/documentation/computers/processors.html>
2. Samsung Electronics, "Exynos 9110: Wearable Processor," *Samsung Semiconductor*, Accessed: Oct. 30, 2024. [Online]. Available: https://semiconductor.samsung.com/processor/wearable-processor/exynos-9110/
3. Samsung Electronics, "Exynos W1000: Wearable Processor," *Samsung Semiconductor*, Accessed: Oct. 30, 2024. [Online]. Available: <https://semiconductor.samsung.com/processor/wearable-processor/exynos-w1000/>

# Patent rights declarations(s)

**Fraunhofer HHI may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_