HeartKey® I²C Interface for the Maxim Sensor Hub MCU

Table of Contents

HeartKey® Algorithms

Introduction

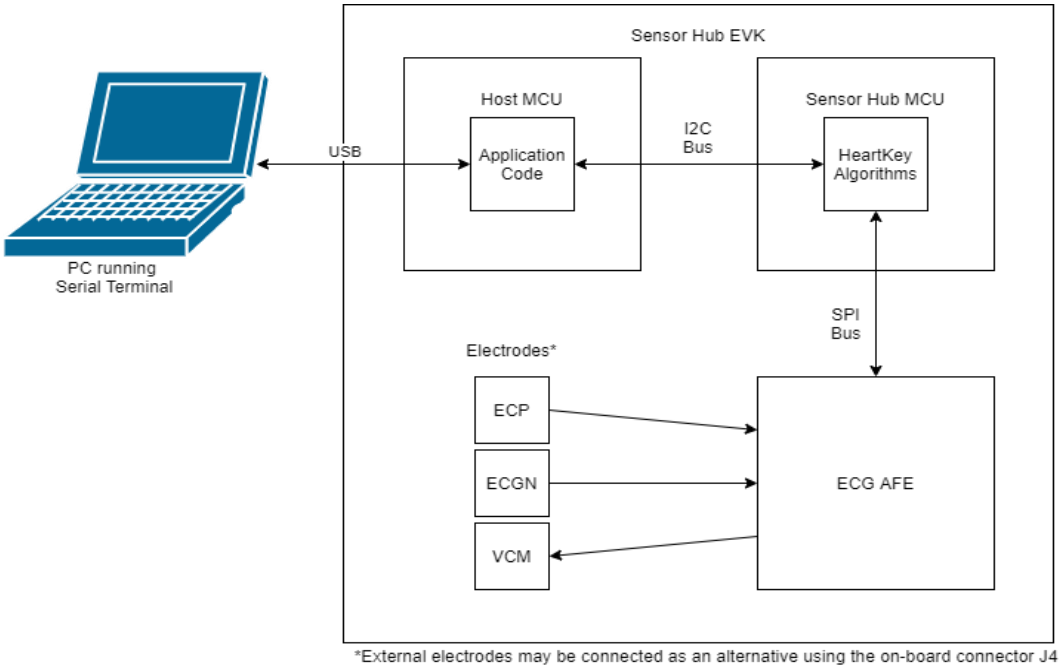
The HeartKey® by B-Secur brings world-class ECG algorithms for a competitive edge to end products. Designed with dry electrodes in mind, the HeartKey®'s signal conditioning technology provides medical grade ECG accuracy for health and wellness analysis and biometric Identification in real time.

Purpose and Scope

This document provides informs the evaluators of the Maxim Sensor Hub on how to use the Sensor Hub demo board and the embedded HeartKey® algorithms. It also informs the integrators of the Maxim Sensor Hub on how to use the pre-embedded HeartKey® algorithms. It details the API, key software modules, and how to use a CLI with the Maxim Sensor Hub demo board. This document does not contain example codes. Software engineers can use the accompanying code as the primary reference of information (see the Key Software Modules section).

System Architecture

The following diagram shows the basic system architecture of the Maxim Sensor Hub development platform with the B-Secur HeartKey® algorithms.



The Host MCU allows the compilation of application code to interface with the HeartKey® algorithms over an I^2C interface. The vital signs data is received through the Electrodes and ECG AFE. It is processed by the HeartKey® on the Sensor Hub. The algorithm outputs are then available to the Host MCU through the I^2C interface.

Quick Start

Warning: *It is essential to isolate the PCB setup from any mains electricity sources to ensure the representative performance of the system. The presence of mains noise can significantly reduce the quality of the ECG signal acquired by the system.*

Even using the system in close physical proximity to mains electricity circuits can cause the coupling of mains noise onto the system. It is advised to distance the system setup from any mains electricity circuit as much as possible

**! Disconnect the PC from the mains supply and run on battery power! **

! Disconnect the PC from any monitor, USB hub, or other peripheral devices powered from the mains supply!

The Sensor Hub EVK comes with a preloaded demo application, which features a Command Line Interface for the easy demonstration and evaluation of the Maxim ECG solution with the HeartKey® algorithms.

- Connect the Sensor Hub EVK board to your PC with a micro USB cable.
- Using a terminal program, set up a connection to the Maxim Featherwing board through the relevant COM port.
- Configure the connection for:
    - Speed: 115200
    - Data: 8 bits
    - Parity: None
    - Stop bits: 1
    - Flow control: None
- Once the board is connected, press the RESET button on the Sensor Hub EVK board (located close to the micro USB port).
- A welcome message displays on the screen in the terminal program followed by a list of available commands.
- Hold the Sensor Hub EVK board as advised in the Holding the board for ECG Analysis section.
- Observe the terminal interface for the ECG analysis output from the HeartKey® algorithms.

Enrolling a User

- Send the command **startenrol** over the terminal connection.
- A feedback in the terminal window says **Enrol Started**.
- Hold the Sensor Hub EVK board as advised in the Holding the board for ECG Analysis section.
- Continue holding the electrodes and observe the enrolment progress (%) until enrollment into the system happens (Enrolment Progress = 100%, Status = Authenticated).

Authenticating a User

Assuming a user is already enrolled:

- Send the command **startauth** over the terminal connection.
- A feedback in the terminal window says **Auth Started**.
- Hold the Sensor Hub EVK board as advised in the Holding the board for ECG Analysis section.
- The status feedback says **Authenticated** if the Authentication is successful. It says **Not Authed** if failed and provides a rejection reason.

Holding the Board for ECG Analysis

The following is recommended to provide a good quality ECG to the Sensor Hub EVK:

- Hold the board with the daughter board facing towards you and the micro USB port pointing downwards.
- Place the left thumb on the ECGP electrode.
- Place the right thumb on the ECGN electrode.
- Place the right pointer finger on the opposite side of the board on the VCM electrode.
- Rest the arms on a stable surface.
- Remain as still as possible.
- Do not slide the thumbs/fingers on the electrode pads.
- Relax and do not squeeze the electrode pads, maintaining a stable connection between the pads and fingers.

LED Behavior

LED Behavior	Meaning
Quick red flash x 3	Host MCU boot/reset
Green flash/second	Main loop running
Blue flash	HeartKey® data update
Red flash	I_2C communication error

Key Software Modules

I²C HeartKey® Data Module - bs_i2c_hk_data.c

This is a primary reference for the application developer as it contains the API and data structure information for the output data from the HeartKey® algorithms on the Sensor Hub MCU. The developer can easily read data from the **hk_data_g** structure to receive the latest data retrieved by the HeartKey® algorithms.

B-Secur Data Types - bsec_types.h

The developer can also use this library header as a useful reference to interpret the data received from the HeartKey® algorithms.

I²C Module - bs_i2cm.c

Contains general functions for the I²C interface between the Host MCU and Sensor Hub MCU.

USB Module - bs_usb.c

Contains general functions for the Host MCU USB interface.

Initializing the Module

The application code must initialize several software modules before attempting to communicate to the Sensor MCU or a connected PC.

I²C MCU Interface

- bs_i2cm_init()
  Initializes the I²C bus between the Sensor MCU and Host MCU.
- bs_i2c_hk_data_init()

Initializes the data structures that contain the data retrieved from the HeartKey® algorithms.

Once initialized, the I²C communication link can be checked with an echo request using the **bs_i2c_hk_data_send_echo_request()** API.

USB Terminal Interface

- bs_usb_init()
  Initializes the USB interfacfe on the Host MCU to communicate with a connected PC.
- hk_term_init()
  Initializes the module that transmits data retrieved from the HeartKey® algorithms.
- bsec_cli_init()
  Initializes the module that relays input commands from the PC to the Sensor Hub MCU.

Graceful Shutdown

The Sensor Hub must be shutdown using the **Shutdown Sensor Hub** API to ensure no loss of data.

HeartKey® Data API Information

API	I²C	Demo CLI
Get HR Info	bs_i2c_hk_data_get_heart_rate	NA*
Get HRV Info	bs_i2c_hk_data_get_hrv_info	NA*
Get Stress Info	bs_i2c_hk_data_get_stress_info	NA*
Get EE Info	bs_i2c_hk_data_get_energy_expend_info	NA*
Get User ID Info	bs_i2c_hk_data_get_user_id_info	NA*
Get User Presence Info	bs_i2c_hk_data_get_user_presence	NA*
Enter Boot Mode	bs_i2c_hk_data_switch_sensor_to_boot_mode	sensorboot
Start Enrol	bs_i2c_hk_data_start_enrol	startenrol
Unenrol User	bs_i2c_hk_data_unenrol	unenrol
Start Authentication	bs_i2c_hk_data_start_auth	startauth
Alternative Authentication	bs_i2c_hk_data_alternative_auth_request	altauth
Add User Template	bs_i2c_hk_data_add_user_template	addtemplate
Set Library Mode	bs_i2c_hk_data_set_lib_mode	setlibmode
Set User Metadata	bs_i2c_hk_set_user_metadata	setusermeta
Get User Metadata	bs_i2c_hk_get_user_metadata	getusermeta
Shutdown Sensor Hub	bs_i2c_hk_data_sensor_shutdown_request	sensorshutdown

*These APIs do not have CLI commands as the data retrieved by them is done so

autonomously over the CLI.

HeartKey® Algorithms

There are several algorithms available to the application code developer from the B-Secur HeartKey® algorithm suite.

User Presence

The HeartKey® reports the status of a person by analyzing an ECG signal. The user presence status can be one of the three states:

- User Not Present
- User Present
- User Alive

The user presence data can be retrieved using the **Get User Presence Info** API.

Health and Wellness

Heart Rate (HR)

The HeartKey® can detect a person's heart rate by analyzing a person's ECG signal. The R-peaks of the ECG signal are identified over several heartbeats to do so. The HeartKey's HR algorithm was validated to medical standards and tested against an FDA cleared medical device. The performance on a dry electrode wrist-worn wearable was equivalent to the wet electrode medical device. This algorithm is pending FDA clearance.

The heart rate data can be retrieved from the HeartKey® through the **Get HR Info** API.

Estimated Heart Rate

The HeartKey® reports an estimated heart rate by detecting as little as **3 heartbeat (RR) intervals**. This is a quick indication of the heart rate but does not carry out some of the more advanced processing for the accurate heart rate.

Accurate Heart Rate

The HeartKey® reports an accurate heart rate by detecting **9 heartbeat (RR) intervals**.

Heart Rate Variability (HRV)

The HeartKey® can detect a person's heart rate variability by analyzing a person's ECG signal. The Heart Rate Variability is a measure of the small differences in the RR intervals and can indicate many physiological parameters like stress and fatigue.

The HeartKey® reports a person's HRV in milliseconds after a recording of **30 heartbeat (RR) intervals** of sufficient signal quality and *updates every 15 intervals thereafter*.

There are several ways the HRV can be reported. The HeartKey® reports the HRV as **RMSSD** (Root Mean Square of Successive [RR Interval] Differences).

The HRV data can be retrieved from the HeartKey® through the **Get HRV Info** API.

Stress

The HeartKey® can detect a person's physiological stress by analyzing a person's ECG signal. Although it may be obvious a person's stress is higher with a higher heart

rate (e.g., due to exercise), the HeartKey® Stress algorithm detects physiological stress when the body is not exercising. Stress can also be present in a person's body because of cognitive stress, illness, or alcohol consumption.

Stress data can be retrieved from the HeartKey® through the **Get Stress Info** API. The stress algorithm is made more accurate by configuring accurate user metadata using the **Set User Metadata** API.

Stress Figure

The HeartKey® reports a person's stress figure, i.e., a **number between 1 and 100** after a recording of **30 heartbeat (RR) intervals** of sufficient signal quality and *updates every 15 intervals thereafter*. People can have a range of stress scores at the same level of stress. So, monitoring the score over time and looking for consistent abnormal high levels is important to health and wellness.

Stress Level

The HeartKey® reports a person's stress level whenever a stress figure is calculated. The stress level can be **one of four levels: Recovery, Low, Medium, or High**.

Energy Expenditure (EE)

The HeartKey® can measure a person's energy expenditure (or calorie burn) by analyzing their ECG signal. The HeartKey® updates and reports energy expenditure every 60 seconds in relation to two components: Resting and Active. Each component and the total value are reported in calories per minute (kCal/min).

The EE data can be retrieved from the HeartKey® through the **Get EE Info** API. The EE algorithm is made more accurate by configuring accurate user metadata using the **Set User Metadata** API.

Resting Energy Expenditure

The HeartKey® reports the calculation of a person's baseline energy expenditure when at rest. This component is attributable to the bodily functions that occur at rest such as breathing, digestion, etc.

Active Energy Expenditure

The HeartKey® reports the calculation of a person's energy expenditure due to physical activity. This component is attributable to movement and activity such as walking, running, swimming, or other physical exertion.

Total Energy Expenditure

The HeartKey® reports the calculation of a person's total energy expenditure. This is the simple summation of the **Resting** and **Active** components.

User ID

The HeartKey® can identify people by analyzing a person's ECG signal. The whole morphology of the ECG signal (i.e., the PQRST complex) is required to do this. User Identification is not possible just by using the heartbeat intervals (or RR intervals). The HeartKey® must be taught the ECG signature of the person beforehand to identify them. This is called an enrolment.

Enrolment

The enrolment process teaches the ECG signature of a person to the HeartKey®. This process is required for the HeartKey® to identify a person in the future. User identification cannot be done without enrolment. This is the case of any biometric modality. An enrolment is completed by recording a person's ECG signal. This recording is not defined by a length of time, but by a number of heartbeats. An enrolment requires a recording of 30 heartbeats of sufficient signal quality. A Heart Model is created within the HeartKey® in the enrolment process.

An enrolment can be initiated by calling the **Start Enrol** API. A user can be unenrolled by calling the **Unenrol User** API.

Authentication

The authentication process is how the HeartKey® identifies the user by analyzing their ECG signature. Given a user was enrolled into the system on a previous occasion (see the Enrolment section), the user can be identified by the system using the HeartKey®'s authentication process.

Authentication can take as little as a few heartbeats in the verification mode. The HeartKey® applies a timeout of 10 heartbeats in the identification mode, which enrolls multiple users. An identification decision is made after this.

An authentication attempt can be initiated using the **Start Authentication** API. Information about the User ID status of the system (such as if a user was authenticated, or the reason a user was not authenticated, etc.) can be retrieved by calling the **Get User ID Info** API.

Alternative Authentication

The HeartKey® can be deployed in a system with other methods of user authentication. If the user is authenticated by system using some other method, the system can inform the HeartKey® by calling the **Alternative Authentication** API. A typical circumstance where this API is used is when the user is not recognized when presenting their ECG for identification. Calling this API informs the HeartKey® that the user's identity is verified. The HeartKey® then uses any available ECG data received during the user authentication attempt to add a new template for that user.

Adding User Templates

A user template is created upon enrolling a user into the HeartKey®. This template is stored by the HeartKey® to identify that user. More ECG recordings enable a more repeatable and faster authentication for some users. This API can be used to create another template for the user by recording more ECG data. This can be done at will. However, a good use case is where a user is not recognized after an authentication attempt and there is not enough data to automatically add a template by the **Alternative Authentication** API. The HeartKey® can initiate another ECG recording for a user just like an enrolment and add this template to the information recorded for that user by calling the **Add User Template** API.

User ID Modes

The HeartKey® facilitates two identification modes, the verification and identification modes. The configuration of this mode allows the better applicability of the user identification feature to a particular use case. The identification mode can be changed using the **Set Library Mode** API.

Verification

The verification mode discerns if a person matches a single person's identity or not.

Identification

The identification mode discerns the identity of a person that matches within a group of identities. There are two sub-types of the identification mode:

- Closed Set
  The HeartKey® identifies a user from a closed set of up to five users. This setting always identifies as one of the enrolled users, whichever is the closest match. This is useful for applications where the user ID is used as a personalization feature.
- Open Set (Beta)
  The HeartKey® identifies a user from a set of up to five users. It also identifies a user as unrecognized if there is no match with any enrolled user. This setting can be used for a security application where it is important to deny access to a system to an unrecognized user.