Application Note

Self-Learning AI Software: Fitness Tracking Application



Self-Learning AI Software: Fitness Tracking Application

Document revision	3.1
Document release date	August 2026
Document number	BST-BHI385-AN001-01
Sales Part Number	BHI385: 0 273 017 068
Notes	Data and descriptions in this document are subject to change without notice. Product photos and pictures are for illustration purposes only and may differ from the real product appearance. The technical details and legal disclaimer of the respective product data sheet apply.

1 Introduction

This document describes the self-learning AI software library and contains information on performance results for the fitness application area.

It also describes how to integrate the Self-Learning AI algorithm (also referred to as Klio) featured on the BHI385 into any application.

The BHI385 related specifics of enabling a sensor and receiving the data stream are not covered in this document, however, some Self-Learning AI differences are outlined.

Before reading this document, it is recommended that users familiarize themselves with how to initialize and set up streaming data from the BHI385. An example of this can be found in `examples/quaternion` of the BHI385 sensor API root tree. It is also assumed that the integration example in `examples/klio` has been skimmed through.

Throughout this document, paths to the BHI385 sensor API will be denoted as `bhi385/<path>`, for the above-mentioned examples would be referred to as `bhi385/examples/quaternion` and `bhi385/examples/klio`.

Table of Contents

1 Introduction 2

2 Self-learning AI software 6

2.1 Learning 6
2.2 Recognition 7
2.3 Performance results 8
2.3.1 Learning results 9
2.3.2 Recognition results 10

3 System resource usage 13

3.1 Memory measurement 13
3.2 Power measurement 14
3.3 CPU load measurement 16

4 Code example description 17

4.1 Run the example code 17

5 API usage 18

5.1 Klio sensor parameters 18
5.2 Enabling the sensor 18
5.3 Reading a learnt pattern 19
5.4 Uploading a pattern 19
5.5 Similarity score 19
5.6 Algorithm reset 19
5.7 Setting parameters 19
5.8 Klio driver parameters 20
5.8.1 Algorithm version (KLIO_PARAM_ALGORITHM_VERSION) 20
5.8.2 Ignore insignificant movement (KLIO_PARAM_LEARNING_IGNORE_INSIG_MOVEMENT) 20
5.8.3 Recognition responsiveness (KLIO_PARAM_RECOGNITION_RESPONSIVNESS) 20
5.8.4 Pattern block size (KLIO_PARAM_PATTERN_BLOB_SIZE) 20
5.8.5 Recognition max patterns (KLIO_PARAM_RECOGNITION_MAX_PATTERNS) 20
5.9 Self-learning AI sensor notification 21
5.9.1 Callback register 21
5.9.2 Sensor frame example 21
5.10 Driver status 23
5.11 Additional code example notes 23

6 Using Klio with bhy2cli tool 24

6.1 Basic bhy2cli commands 25
6.1.1 Trigger a soft reset to the sensor 25
6.1.2 Reset, and then load the Klio firmware to RAM and boot 25
6.1.3 Show the list of virtual sensors 25
6.1.4 Enable a virtual sensor 27
6.1.5 Enable multiple virtual sensors simultaneously 27
6.2 Klio related bhy2cli commands 29
6.2.1 Get and reset current Klio driver status 31
6.2.2 Set Klio state 31
6.2.3 Get current Klio state 32

6.2.4 Load a pattern for recognition 32
6.2.5 Enable pattern for recognition 32
6.2.6 Disable pattern for recognition 34
6.2.7 Automatically use learnt patterns for recognition 34
6.2.8 Switch pattern between left/right hand 36
6.2.9 Get Klio parameter 36
6.2.10 Set Klio parameter 37
6.2.11 Get Klio pattern parameter 38
6.2.12 Set Klio pattern parameter 38
6.2.13 Get similarity score for two patterns 38
6.2.14 Get similarity score for one or more stored patterns 39

7 References 40

8 Legal disclaimer 41

9 Document history and modification 42

List of figures

Figure 1: Flow chart of the Self-learning AI “Learning” Algorithm 6
Figure 2: Flow chart for the recognition algorithm 7
Figure 3: Axis orientation of IMU sensor 8
Figure 4: Run the klio example code 17
Figure 5: Klio Command Flow 30

List of tables

Table 1: Learning results 9
Table 2: Recognition results 11
Table 3: Performance comparison against reference patterns 12
Table 4: Turbo ON and Algorithm Input ODR 25 Hz 14
Table 5: Turbo ON and Algorithm Input ODR 50 Hz 14
Table 6: Turbo OFF and Algorithm Input ODR 25 Hz 14
Table 7: Turbo OFF and Algorithm Input ODR 50 Hz 15
Table 8: Turbo ON and Algorithm Input ODR 25 Hz 16
Table 9: Turbo ON and Algorithm Input ODR 50 Hz 16
Table 10: Turbo OFF and Algorithm Input ODR 25 Hz 16
Table 11: Turbo OFF and Algorithm Input ODR 50 Hz 16
Table 12: List the Klio sensor parameters 18
Table 13: List the Klio driver parameters 20
Table 14: Klio Sensor Data Format 21
Table 15: Driver Status 23

2 Self-learning AI software

The self-learning AI software can be used to learn new cyclic gestures and provides responsive recognition of previously learnt cyclic gestures together with an instant repetition count, as well as a score indicating how accurately the gesture was performed. The cyclic gesture needs to be the same in between the repetitions and the device needs to come to the original position to close one cycle.

2.1 Learning

The self-learning algorithm is capable to learn human cyclic activities, such as bicep curls, triceps extensions and squats. The speed of the activity to be learned should be in the default range of in the range of 0.3 to 1.8Hz (one repetition in 1/0.3s to one repetition in 1/1.8s). The supported activity speeds or frequencies are configurable. The self-learning algorithm estimates a number of parameters, where the phase and frequency are two of them, from the sensor input signals. A number of so-called estimators, each operating at a specific frequency is employed to estimate the parameters of interest. $1 \times N$ estimators operate at different frequencies to learn the parameters of interest. The last step in the algorithm is a logic to determine what estimator, that provides the correct values of the parameters, has converged to produce the learnt pattern.

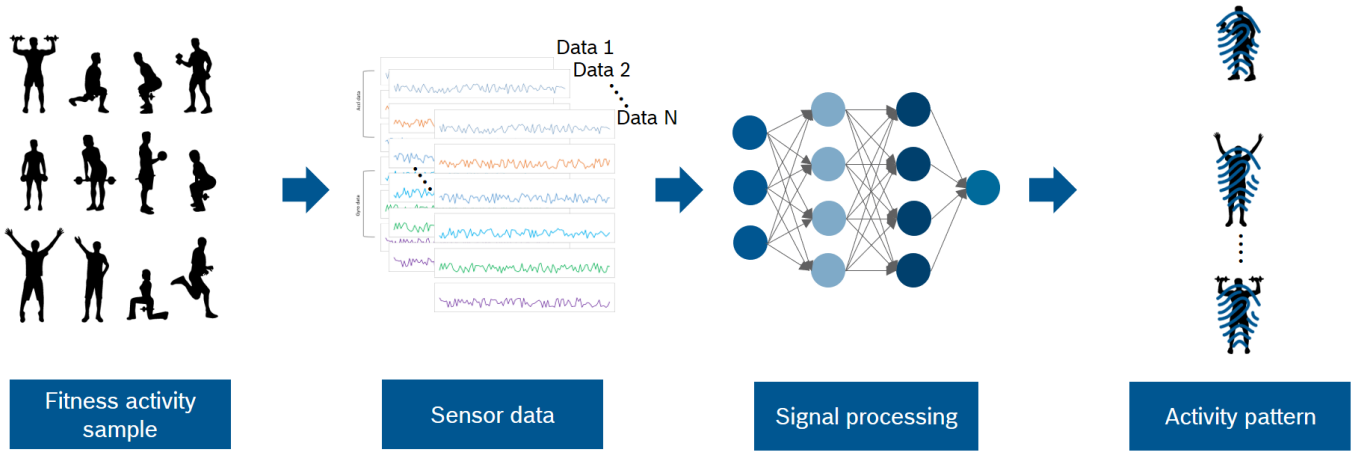


Figure 1: Flow chart of the Self-learning AI “Learning” Algorithm

2.2 Recognition

The recognition algorithm relies on previously learnt patterns and/or built-in patterns to recognize ongoing activities and count repetitions. The recognition algorithm employs a number of detectors, one for each learnt pattern, to determine the most likely ongoing activity and count the repetitions in real-time. Each detector needs to estimate a number of parameters to try to align the pattern to the incoming sensor signals. The logic determines the best matching activity but also takes into account the case with no matching activity at all. Figure 2 shows the flow chart for the recognition algorithm.

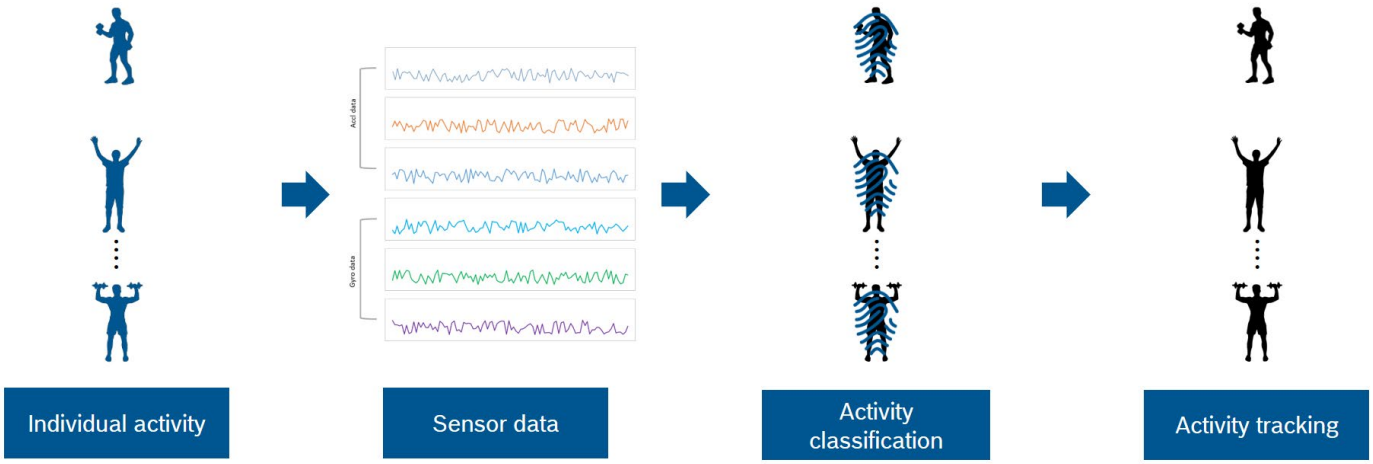


Figure 2: Flow chart for the recognition algorithm

2.3 Performance results

The self-learning AI algorithm has been tested on datasets collected from 10 users performing various weight based and bodyweight-based exercises. The performance results are reported in this section.

Note:

- Multiple variations of a specific exercise can exist. The results reported are based on a specific way (deemed to be more common) of doing the exercises.
- For these exercises, reference patterns can also be made available, based on the same datasets, which can be deployed at an application level – most common being an Android application.
- Bosch Sensortec can be contacted to get more information on the reference exercise patterns and the way these exercises were performed.
- Exercise classification and repetition counting accuracy is provided as F1 scores.
- The datasets were collected using a wrist worn wearable device on the left arm consisting of BHI385.
- The algorithm uses accelerometer and gyroscope. The sampling rates of both sensors are set to 50 Hz.
- An important notice is that the sensor and the device orientation need to be consistent between Learning and Recognition.
- The axis orientation of IMU sensor with respect to the wrist wearable device is shown below:

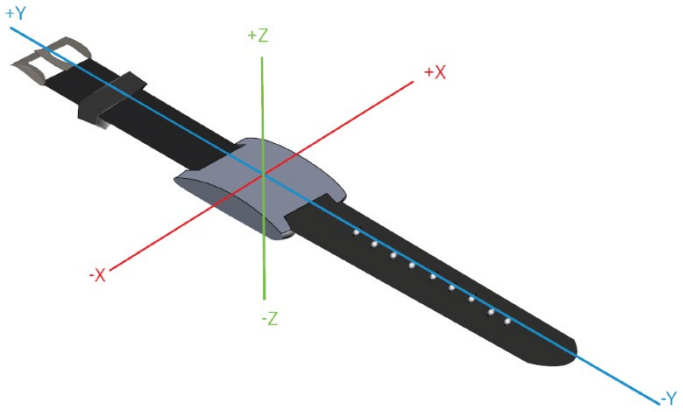


Figure 3: Axis orientation of IMU sensor

2.3.1 Learning results

Table 1: Learning results

				Learning time (s)		
#	**Exercise name**	**Learning rate***	**Number of datasets**	**Mean**	**Median**	**Standard deviation**
1	Dumbbell Bench Press	0.95	55	17.54	16.64	6.61
2	Dumbbell Biceps Curl	1	52	9.61	10.27	2.6
3	Dumbbell Bicep Curl Palm Face Front	1	52	7.83	6.88	1.64
4	Dumbbell Chest Fly	1	48	9.85	10.13	1.72
5	Crunches	1	63	8	7.88	1.58
6	Jumping Jacks	0.95	61	6.9	6.86	1.04
7	Kettlebell Swing	1	49	7.63	7.8	0.99
8	Lower Back Extension	1	61	9.23	10.06	1.99
9	Dumbbell Pull Over	0.94	48	11.37	10.28	5.84
10	Row Machine	0.94	62	18.81	13.57	17.18
11	Russian Twist	1	56	6.7	6.4	0.94
12	Side Lateral Raise	1	43	7.58	6.84	1.85
13	Bodyweight Squats	1	60	7.91	7.18	1.54
14	Dumbbell Triceps Extensions	1	49	8.14	7.64	1.87
15	Dumbbell Triceps Kickbacks	1	50	7.49	6.77	2.03

*Learning rate measures if learning was possible on the considered datasets.

The most difficult exercises to learn are bench press and row machine. It takes around 16s to learn a new bench press exercise and around 14s to learn a row machine. The spread in learning time is quite big for row machine, indicated by the large standard deviation and the displacement of the median and mean.

Learning time is also dependent on the complexity of the exercise being performed including with or without weights.

2.3.2 Recognition results

2.3.2.1 F1 score calculation procedure

Definition: https://en.wikipedia.org/wiki/F1_score

TP - Sum of all repetitions that are counted correctly for current exercise of interest
TT - Sum of all "false counts" more counts than actual true repetitions
FN - All the missing repetitions for the current exercise that are predicted incorrectly and ends up in false negatives.
FP - Sum of all repetitions that are counted incorrectly, i.e. false matches (counting repetition on wrong exercises).

β = 1 for F_1 score
β = 0.5 for $F_{0.5}$ score

$$Precision = \frac{TP}{TP + FP + TT}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_\beta = \frac{(1 + \beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Precall}$$

2.3.2.2 Results

For this case only learnt exercise patterns from the user of interest is used in the recognition. The F1 score is checked for each combination of datasets for a specific user and exercise. The same dataset is never used for both learning and recognition. The performance for a specific exercise is checked when the full number of patterns are present in the recognition. However, all other patterns (not for the exercise of interest) are a combination of the user's own patterns for each exercise. It is needed to prevent a combinatorial explosion by checking all combination of datasets for every exercise. The number of combinations will be in the order of (number of permutations for each exercise) to the power of (total number of exercises). This simplification should be similar to randomly selecting permutations and evaluating performance. The combined patterns are created without any weight.

Table 2: Recognition results

#	Exercise name	F1 score										
		Combined	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10
1	Dumbbell Bench Press	0.94	0.95	0.98	0.89	0.89	0.99	0.96	0.97	0.90	0.93	0.98
2	Dumbbell Biceps Curl	0.95	0.94	0.89	0.99	0.97	0.98	0.57*	0.99	0.91	0.99	1.00
3	Dumbbell Bicep Curl Palm Face Front*	0.97	0.99	0.90	0.97	0.99	0.97	0.97	0.99	0.94	1.00	0.98
4	Dumbbell Chest Fly	0.97	0.99	0.98	0.99	0.98	0.99	0.86	0.99	0.99	0.98	0.99
5	Crunches	0.97	0.99	0.98	0.99	0.95	0.98	0.97	0.99	0.97	0.97	0.93
6	Jumping Jacks*	0.96	0.99	0.98	0.98	0.96	0.86	0.95	0.96	0.97	0.97	0.98
7	Kettlebell Swing	0.96	0.99	0.91	0.99	0.91	0.94	0.99	0.97	0.98	0.93	0.99
8	Lower Back Extension	0.96	0.99	0.92	0.97	0.90	0.94	0.98	1.00	0.98	0.94	0.99
9	Dumbbell Pull Over	0.96	0.99	0.91	0.91	0.98	0.99	0.94	0.97	0.99	0.98	0.97
10	Row Machine	0.95	0.99	0.99	0.81	0.95	0.98	0.96	0.91	0.99	0.92	1.00
11	Russian Twist	0.95	0.63*	0.99	0.98	0.99	0.99	0.98	0.98	0.96	1.00	0.99
12	Side Lateral Raise	0.97	0.98	0.98	0.98	0.97	0.93	0.98	0.99	0.94	0.98	0.99
13	Bodyweight Squats	0.97	0.97	0.96	0.98	0.92	0.98	0.94	0.99	0.96	0.98	0.98
14	Dumbbell Triceps Extensions	0.97	0.99	0.93	0.94	0.97	0.98	0.96	0.98	0.98	0.99	0.95
15	Dumbbell Triceps Kickbacks	0.95	0.98	0.98	0.90	0.98	0.95	0.99	0.98	0.97	0.84	0.97

* "Russian Twist", User 1 and "Dumbbell Biceps Curl", User 6 show poor performance even using their own learnt patterns. It indicates that they are not consistent in the techniques or speed.

Overall accuracy of recognition classification and repetition counting improves when users are consistent in performing the exercise. Recognition classification accuracy can also be improved by including exercises, which are more different from the sensor perspective. For example, Crunches and Dumbbell pullover have shown higher correlation in the datasets and can be grouped separately to avoid False matches.

2.3.2.3 Performance comparison with reference patterns V2

Bosch Sensortec can also make available “Reference patterns” for these exercises. These reference patterns are generated by combining several users’ data / patterns after carefully considering and optimizing the performance for the overall dataset.

Reference Patterns V2: Patterns are now adaptive. This allows the user to perform the exercise with a less strict technique and the reference pattern will adapt seamlessly to the user‘s own technique.

Because of limited dataset, results are to be seen more as an indicator.

Table 3: Performance comparison against reference patterns

#	Exercise name	F1 score
		Reference pattern V2
1	Dumbbell Bench Press	0.94
2	Dumbbell Biceps Curl	0.94
3	Dumbbell Bicep Curl Palm Face Front	0.93
4	Dumbbell Chest Fly	0.94
5	Crunches	0.96
6	Jumping Jacks	0.96
7	Kettlebell Swing	0.99
8	Lower Back Extension	0.94
9	Dumbbell Pull Over	0.94
10	Row Machine	0.93
11	Russian Twist	0.95
12	Side Lateral Raise	0.96
13	Bodyweight Squats	0.96
14	Dumbbell Triceps Extensions	0.93
15	Dumbbell Triceps Kickbacks	0.97

3 System resource usage

The BHI385 is an ultra-low power, customizable smart sensor consisting of Bosch Sensortec's new, programmable 32-bit microcontroller (Fuser2), a state-of-the-art 6-axis IMU (3-axis Accelerometer + 3-axis Gyroscope) and a powerful event-driven software framework containing pre-installed sensor fusion and other sensor data processing software in a small 44-pad LGA package.

The Fuser2 Core can be configured to operate at 20 MHz (Long Run mode) or 50 MHz (Turbo mode). It can boot from a wide variety of hosts, ranging from a small Cortex-M0™ MCU to multicore application processors, while it also has the ability to run standalone, when booting from an attached flash memory.

The core algorithm library has been deployed with the BHI385. Measurements for Power, Memory and CPU load are provided based on this setup. If the standalone core algorithm library is deployed with any other sensor or MCU combination the measurements will change.

The measurements that running the "Self-Learning AI" functionality are provided for both Long Run mode (Turbo OFF) and Turbo ON.

3.1 Memory measurement

The memory measurement on the BHI385 firmware shows the memory usage of 53.2KB (code and data) for a GCC compiled firmware.

Depending on the target and compiler, this number might vary. The memory usage can also be reduced if the number of patterns to be compared for recognition is reduced from 25.

3.2 Power measurement

Voltage – 1.8V. Fuser2 is not in deep sleep.

Table 4: Turbo ON and Algorithm Input ODR 25 Hz

	CPU Power Measurement	
	VDDIO (µA)	VDD (µA)
Learning - 1 pattern	191	668
Recognition - 1 pattern against 10 patterns	66	668
Recognition - 1 pattern against 10 adaptive patterns	72	668
Recognition - 1 pattern against 15 patterns	80	668
Recognition - 1 pattern against 15 adaptive patterns	87	668
Learning - 1 pattern and Recognition - 1 pattern against 10 patterns	222	668
Learning - 1 pattern and Recognition - 1 pattern against 10 adaptive patterns	228	668
Learning - 1 pattern and Recognition - 1 pattern against 15 patterns	236	668
Learning - 1 pattern and Recognition - 1 pattern against 15 adaptive patterns	244	668
Klio (Learning & Recognition) disabled	19.2	4.9

Table 5: Turbo ON and Algorithm Input ODR 50 Hz

	CPU Power Measurement	
	VDDIO (µA)	VDD (µA)
Learning - 1 pattern	361	668
Recognition - 1 pattern against 10 patterns	111	668
Recognition - 1 pattern against 10 adaptive patterns	125	668
Recognition - 1 pattern against 15 patterns	139	668
Recognition - 1 pattern against 15 adaptive patterns	156	668
Learning - 1 pattern and Recognition - 1 pattern against 10 patterns	423	668
Learning - 1 pattern and Recognition - 1 pattern against 10 adaptive patterns	437	668
Learning - 1 pattern and Recognition - 1 pattern against 15 patterns	451	668
Learning - 1 pattern and Recognition - 1 pattern against 15 adaptive patterns	468	668
Klio (Learning & Recognition) disabled	19.2	4.9

Table 6: Turbo OFF and Algorithm Input ODR 25 Hz

	CPU Power Measurement	
	VDDIO (µA)	VDD (µA)
Learning - 1 pattern	145	668
Recognition - 1 pattern against 10 patterns	47	668
Recognition - 1 pattern against 10 adaptive patterns	51	668
Recognition - 1 pattern against 15 patterns	58	668
Recognition - 1 pattern against 15 adaptive patterns	64	668
Learning - 1 pattern and Recognition - 1 pattern against 10 patterns	169	668
Learning - 1 pattern and Recognition - 1 pattern against 10 adaptive patterns	173	668
Learning - 1 pattern and Recognition - 1 pattern against 15 patterns	180	668
Learning - 1 pattern and Recognition - 1 pattern against 15 adaptive patterns	186	668
Klio (Learning & Recognition) disabled	10.2	4.9

Table 7: Turbo OFF and Algorithm Input ODR 50 Hz

	CPU Power Measurement	
	VDDIO (µA)	VDD (µA)
Learning - 1 pattern	278	668
Recognition - 1 pattern against 10 patterns	81	668
Recognition - 1 pattern against 10 adaptive patterns	91	668
Recognition - 1 pattern against 15 patterns	103	668
Recognition - 1 pattern against 15 adaptive patterns	116	668
Learning - 1 pattern and Recognition - 1 pattern against 10 patterns	326	668
Learning - 1 pattern and Recognition - 1 pattern against 10 adaptive patterns	337	668
Learning - 1 pattern and Recognition - 1 pattern against 15 patterns	349	668
Learning - 1 pattern and Recognition - 1 pattern against 15 adaptive patterns	360	668
Klio (Learning & Recognition) disabled	10.2	4.9

3.3 CPU load measurement

Table 8: Turbo ON and Algorithm Input ODR 25 Hz

	CPU load (%)
Learning - 1 pattern	6.16
Recognition - 1 pattern against 10 patterns	2.09
Recognition - 1 pattern against 10 adaptive patterns	2.49
Recognition - 1 pattern against 15 patterns	2.22
Recognition - 1 pattern against 15 adaptive patterns	3.12
Learning - 1 pattern and Recognition - 1 pattern against 10 patterns	7.44
Learning - 1 pattern and Recognition - 1 pattern against 10 adaptive patterns	7.48
Learning - 1 pattern and Recognition - 1 pattern against 15 patterns	8.02
Learning - 1 pattern and Recognition - 1 pattern against 15 adaptive patterns	8.16

Table 9: Turbo ON and Algorithm Input ODR 50 Hz

	CPU load (%)
Learning - 1 pattern	12.49
Recognition - 1 pattern against 10 patterns	3.54
Recognition - 1 pattern against 10 adaptive patterns	3.89
Recognition - 1 pattern against 15 patterns	4.56
Recognition - 1 pattern against 15 adaptive patterns	5.32
Learning - 1 pattern and Recognition - 1 pattern against 10 patterns	14.46
Learning - 1 pattern and Recognition - 1 pattern against 10 adaptive patterns	14.98
Learning - 1 pattern and Recognition - 1 pattern against 15 patterns	15.69
Learning - 1 pattern and Recognition - 1 pattern against 15 adaptive patterns	16.18

Table 10: Turbo OFF and Algorithm Input ODR 25 Hz

	CPU load (%)
Learning - 1 pattern	15.85
Recognition - 1 pattern against 10 patterns	4.44
Recognition - 1 pattern against 10 adaptive patterns	5.26
Recognition - 1 pattern against 15 patterns	5.48
Recognition - 1 pattern against 15 adaptive patterns	6.62
Learning - 1 pattern and Recognition - 1 pattern against 10 patterns	18.19
Learning - 1 pattern and Recognition - 1 pattern against 10 adaptive patterns	18.95
Learning - 1 pattern and Recognition - 1 pattern against 15 patterns	19.24
Learning - 1 pattern and Recognition - 1 pattern against 15 adaptive patterns	19.8

Table 11: Turbo OFF and Algorithm Input ODR 50 Hz

	CPU load (%)
Learning - 1 pattern	30.37
Recognition - 1 pattern against 10 patterns	8.74
Recognition - 1 pattern against 10 adaptive patterns	9.83
Recognition - 1 pattern against 15 patterns	10.78
Recognition - 1 pattern against 15 adaptive patterns	12.43
Learning - 1 pattern and Recognition - 1 pattern against 10 patterns	36.05
Learning - 1 pattern and Recognition - 1 pattern against 10 adaptive patterns	37.09
Learning - 1 pattern and Recognition - 1 pattern against 15 patterns	38.29
Learning - 1 pattern and Recognition - 1 pattern against 15 adaptive patterns	39.92

4 Code example description

The BHI385 sensor API contains an integration example of the Self-learning AI software, `bhi385/examples/klio`. It can be downloaded via the link: https://github.com/boschsensortec/BHI385_SensorAPI/releases. With installing the COINES SDK, copy the BHI385 sensor API folder to COINES/vx.x.x/examples.

The example contains one pre-learnt pattern which can be recognized by keeping the device level with the BHI385 facing upward and moving the device up and down at about 1.5Hz. The example further enables both learning and recognition and writes back the patterns learnt and enables them provided that no similar pattern is already present. As such, it demonstrates the core functionality of the software.

To learn a pattern, perform a repetitive motion (such as moving the device in a figure eight, or side to side etc) with a relatively consistent speed. The learning progress indicator will show how far the learning procedure has come. The learnt pattern will be written back and enabled for recognition if a similar pattern is not already present. Note that learning should be enabled on demand in a real application and not always active as it is in the example.

Upon successful learning, you should get a “Pattern learnt” notification followed by recognition counts on a new pattern index (patterns added during runtime start at index 1 in the example).

When a pattern is written back to the BHI385, the recognition counts are reset. This should not be a problem in practice since learning should be run on demand.

Open the terminal with folder *‘COINES/vx.x.x/examples/BHI385_SensorAPI/examples/klio*’ and run the building command ‘*mingw32-make*’ after the complier has been successfully installed. The detailed information on how to setup the building environment can refer to BHI360/BHI380/BHI385 SDK Quick Start Guide document.

4.1 Run the example code

```
> .\klio.exe
Host Interface : SPI
Chip ID read 0x7C
Host interrupt control
    Wake up FIFO enabled.
    Non wake up FIFO enabled.
    Status FIFO disabled.
    Debugging disabled.
    Fault enabled.
    Interrupt is active high.
    Interrupt is level triggered.
    Interrupt pin drive is push-pull.
Loading firmware into RAM.
Booting from RAM.
Boot successful. Kernel version 2380.
[META EVENT WAKE UP] Firmware initialized. Firmware version 5991
[META EVENT] Firmware initialized. Firmware version 5991
Enable Klio at 25.00Hz.
SID: 112; T: 3.489812500; Learning [Id:-1 Progress:20 Change:0]; Recognition[Id:255 Count:0.000000 Score:0.000000]
SID: 112; T: 4.137453125; Learning [Id:-1 Progress:40 Change:0]; Recognition[Id:255 Count:0.000000 Score:0.000000]
SID: 112; T: 4.258906250; Learning [Id:-1 Progress:40 Change:2]; Recognition[Id:255 Count:0.000000 Score:0.000000]
SID: 112; T: 4.339859375; Learning [Id:-1 Progress:20 Change:2]; Recognition[Id:255 Count:0.000000 Score:0.000000]
SID: 112; T: 4.501781250; Learning [Id:-1 Progress:40 Change:0]; Recognition[Id:255 Count:0.000000 Score:0.000000]
SID: 112; T: 4.947031250; Learning [Id:-1 Progress:60 Change:0]; Recognition[Id:255 Count:0.000000 Score:0.000000]
SID: 112; T: 5.392234375; Learning [Id:-1 Progress:60 Change:2]; Recognition[Id:255 Count:0.000000 Score:0.000000]
SID: 112; T: 5.837484375; Learning [Id:-1 Progress:80 Change:0]; Recognition[Id:255 Count:0.000000 Score:0.000000]
SID: 112; T: 6.242234375; Learning [Id:-1 Progress:80 Change:2]; Recognition[Id:255 Count:0.000000 Score:0.000000]
SID: 112; T: 6.687421875; Learning [Id:5 Progress:100 Change:0]; Recognition[Id:255 Count:0.000000 Score:0.000000]
```

Figure 4: Run the klio example code

5 API usage

This section briefly explains the most important parts of the usage of the APIs.

5.1 Klio sensor parameters

The Klio sensor parameters are defined in `bhi385/bhi385_klio_param_defs.h`. A detailed description can be found in the section on Self-Learning AI Software Parameters in the BHI385 Datasheet.

Table 12: List the Klio sensor parameters

Parameter ID	Parameter Name	Parameter Description	R/W Access
0	KLIO_HIF_PARAM_ALGORITHM_STATE	Set and get the Klio algorithm state	R/W
1	KLIO_HIF_PARAM_PATTERN	Read and write the pattern for Klio algorithm	R/W
2	KLIO_HIF_PARAM_ALGO_DRIVER_PARAMETER	Read and write the driver parameters, detailed in Table 13	R/W
4	KLIO_HIF_PARAM_PATTERN_STATE	Set and get the Klio pattern state. It supports: 0: Disable a pattern, defined as KLIO_PATTERN_STATE_DISABLE 1: Enable a pattern, defined as KLIO_PATTERN_STATE_ENABLE 2: Switch the hand for a pattern(0-right,1-left) KLIO_PATTERN_STATE_SWITCH_HAND 3: Disable a adaptive pattern, defined as KLIO_PATTERN_STATE_AP_DISABLE	Write Only
5	KLIO_HIF_PARAM_PATTERN_SIMILARITY	Perform the pattern similarity	R/W
7	KLIO_HIF_PARAM_DRIVER_STATUS	Read driver status	Read Only
8	KLIO_HIF_PARAM_RESET	Reset the driver and Klio algorithm state	Write Only
9	KLIO_HIF_PARAM_PATTERN_PARAM	Set and get the pattern parameter for Klio algorithm	R/W

5.2 Enabling the sensor

The algorithm state is set by `bhi385_klio_param_set_state` with writing parameter KLIO_HIF_PARAM_ALGORITHM_STATE in Table 12. This function allows you to set whether to enable learning or recognition or reset either of them to their initial states. The sensor also needs to be enabled with an appropriate call to `bhi385_virtual_sensor_conf_param_set_cfg`. In the example, learning and recognition are enabled at start. In a real application, learning should be enabled on demand.

5.3 Reading a learnt pattern

When a new pattern is learnt while the learning index from Klio sensor data is not -1, the new learnt pattern can be read out by `bhi385_klio_param_read_pattern` with reading parameter KLIO_HIF_PARAM_PATTERN in Table 12. The example code shows how to read a new learnt pattern. An example output is provided below.

```
[D]SID: 112; T: 21.941250000; Pattern learnt:
524231060334dca2400ad7233c91199740d8128f401c55f4be11a6b43c46a5b6be770f893dd1b81dbfda
6daa3fd3b8adbef76a20bff6ee94beee47cc3cb02c9f40fee826c1026425400a1d983ff2d6913ee0e9a6
3e7206193eb9f8babc1f8f143ee52634beb31c4d3eed77473ca68004beab0f0e3fab915540ee5b163eab
97bebe189aed3c70378e3cd090013e0310043fe3f4193da6de10be14a8e83a
```

5.4 Uploading a pattern

A pattern is uploaded using the function `bhi385_klio_param_write_pattern` with writing parameter KLIO_HIF_PARAM_PATTERN in Table 12. See the code example for information on how to upload a pattern. The hex string output when a pattern has been learnt needs to be converted to the corresponding binary representation before writing. After the pattern has been written, it needs to be enabled. Use the function `bhi385_klio_param_set_pattern_states` by writing the parameter KLIO_HIF_PARAM_PATTERN_STATE to enable the pattern for recognition. The pattern is enabled for recognition only if it is not already present. See the code example to learn more.

5.5 Similarity score

This feature allows to compute the similarity score for two or more than two patterns with KLIO_HIF_PARAM_PATTERN_SIMILARITY in Table 12. This is done using the API call `bhi385_klio_param_similarity_score` or `bhi385_klio_param_similarity_score_multiple`.

For `bhi385_klio_param_similarity_score`, pattern strings are passed as arguments, while for `bhi385_klio_param_similarity_score_multiple`, pattern ids are passed as arguments. Also, `bhi385_klio_param_similarity_score_multiple` can be used to compute the similarity score between more than two patterns. Refer to the code example and the Doxygen for information on how to use this API.

5.6 Algorithm reset

This feature allows to reset the Klio driver and algorithm. It is implemented using the API call `bhi385_klio_param_reset` with writing the KLIO_HIF_PARAM_RESET in Table 12.

5.7 Pattern Parameter

This feature allows to configure the parameters of individual pattern which contains the exponent scaling factor of a pattern. It's implemented in `bhi385_klio_param_get_pattern_parameter` and `bhi385_klio_param_set_pattern_parameter` with reading/writing the parameter KLIO_HIF_PARAM_PATTERN_PARAM in Table 12 with the patten parameter ID KLIO_PATTERN_PARAM_EXPONENT_SCALING_FACTOR (see `bhi385/bhi385_klio_param_defs.h`).

5.8 Setting parameters

To change parameters, disable the sensor with a call to `bhi385_set_virt_sensor_cfg` (e.g. setting the ODR to 0.0), reset learning/recognition state with `bhi385_klio_param_set_state`, and then make the appropriate call to `bhi385_klio_param_set_parameter` which has the access to KLIO_HIF_PARAM_ALGO_DRIVER_PARAMETER in Table 12.

In addition, you can change parameters of individual patterns. To do this, disable the sensor with a call to `bhi385_virtual_sensor_conf_param_set_cfg` (e.g. setting the ODR to 0.0), reset learning/recognition state

with `bhi385_klio_param_set_state`, load patterns with `bhi385_klio_param_write_pattern`, and then make the appropriate call to enable the pattern with `bhi385_klio_param_set_pattern_parameter`.

5.9 Klio driver parameters

List the supported parameters from Klio driver.

Table 13: List the Klio driver parameters

Parameter ID	Parameter Name	Parameter Description	R/W Access
0	KLIO_PARAM_ALGORITHM_VERSION	Algorithm version	Read Only
2	KLIO_PARAM_RECOGNITION_RESPONSIVNESS	Approximate number of cycles for recognition to recognize an activity	R/W
7	KLIO_PARAM_PATTERN_BLOB_SIZE	Max pattern blob size in bytes	Read Only
8	KLIO_PARAM_RECOGNITION_MAX_PATTERNS	Maximum number of patterns	Read Only
10	KLIO_PARAM_LEARNING_IGNORE_INSIG_MOVEMENT	Flag if insignificant movements should be ignored in learning. Default is 0.	R/W

5.9.1 Algorithm version (KLIO_PARAM_ALGORITHM_VERSION)

This parameter returns the version number of Klio algorithm.

5.9.2 Ignore insignificant movement (KLIO_PARAM_LEARNING_IGNORE_INSIG_MOVEMENT)

This feature prevents learning of small movements. The default value is 0. The parameter id is 10 (see `bhi385/bhi385_klio_param_defs.h`).

5.9.3 Recognition responsiveness (KLIO_PARAM_RECOGNITION_RESPONSIVNESS)

When this feature is increased, the recognition algorithm takes more time to determine the correct ongoing activity (with greater accuracy). In this case, the repetition count is lower than the actual repetitions being performed. When this feature is decreased, the recognition algorithm takes less time to determine an ongoing activity (with lower accuracy). The repetition count will be more similar to the actual repetitions being performed. However, the false match performance will increase to some extent. The default value is 0.1. The parameter id is 2 (see `bhi385/bhi385_klio_param_defs.h`).

5.9.4 Pattern block size (KLIO_PARAM_PATTERN_BLOB_SIZE)

This parameter returns the maximum size of a single pattern blob in bytes.

5.9.5 Recognition max patterns (KLIO_PARAM_RECOGNITION_MAX_PATTERNS)

This parameter returns the maximum number of patterns for Klio algorithm to recognition.

5.10 Self-learning AI sensor notification

5.10.1 Callback register

The sensor will notify the application when it has new data (at the lowest level, the BHI385 will raise an interrupt). The interrupt is handled by registering a callback function with the Self-Learning AI/Klio sensor ID. An example of callback registration is provided below. For details, refer to the code example.

```
bhi385_register_fifo_parse_callback(BHI385_KLIO_SENSOR_ID, parse_klio, (void*)&klio_rt,
&bhy);
```

The `parse_klio` function is called upon notification and parses the sensor frame which contains information on whether a pattern has been learnt or recognized. If a pattern has been learnt, the function `parse_klio_handle_learnt_pattern` in the example reads out the learnt pattern using the function `bhi385_klio_param_read_pattern`. The learnt pattern is then written back and enabled if it is determined that a similar pattern has not been stored (refer to section 5.2 and 5.3). Once the learnt pattern is enabled, it is compared with the existing pattern to compute the similarity score. Refer to section 5.5.3.
As mentioned above, the sensor frame contains information on whether a pattern has been learnt or recognized.

5.10.2 Sensor frame example

As mentioned above, the sensor frame contains information on whether a pattern has been learnt or recognized.

Table 14: Klio Sensor Data Format

Byte Number	Mode	Content	Format
0		Sensor ID	8 bit unsigned, ID:112 for self-learning AI
1	Learning	Reserved	
2		Index	8 bits unsigned
3		Progress	8 bits unsigned
4		Change Reason	8 bits unsigned
5	Recognition	Reserved	
6		Index	8 bits unsigned
7 .. 10		Count	IEEE754 single precision float
11 .. 14		Score	IEEE754 single precision float

Field	Description	
Learning Index	A value of -1 means no new learning has occurred. If the value is >= 0, then a new pattern has been learnt, and reading of this pattern may be performed.	
Learning Progress	While learning a new pattern, this field counts from 0 to 100. When 100 is reached a new pattern will be learnt. If learning is interrupted, this progress will return to 0, and the change reason will be set to indicate why learning is interrupted.	
Learning Change Reason	0	Learning is in progress.
	1	Learning is interrupted by a non-repetitive activity.
	2	Learning is interrupted because no significant movement is detected.

Recognition Index	The index of the recognized activity. 255 means no activity is recognized.
Recognition Count	The current repetition count of the recognized activity.
Recognition Score	The current score of the recognized activity.

An example printout is provided below. See the `bhi385_event_data_klio_t` struct in `bhi385/bhi385_event_data_defs.h` for the definition. Also check out structs `bhi385_event_data_klio_learning_t` and `bhi385_event_data_klio_recognition_t` which constitute the `bhi385_event_data_klio_t`.

An example printout is provided below.

```
[D]SID: 112; T: 30.556; Learning [Id:-1 Progress:100 Change:0]; Recognition[Id:0 Count:2.483 Score:0.952]
```

The Klio sensor ID is 112, no pattern has been learnt (ID: -1) and pattern 0 is recognized with a repetition count of about 2.5 and score of about 95%. Learning progress counts from 0 to 100. Recognition score is between 0.0 and 1.0. See the table above for a description of learning change.

5.11 Driver status

The driver status can be read out after each operation, which provides useful debugging information as shown in table below. See the helper function `print_klio_status` in the example for information on how to use this function. The `print_klio_status` function uses the `bhi385_klio_param_read_reset_driver_status` function in conjunction with the error handler function `print_klio_error` to provide the debugging info.

Table 15: Driver Status

Field Name	Byte Offset	Description
Parameter ID	0x00 - 0x01	0x1907
Length	0x02 - 0x03	Number of bytes to follow = 4
Status	0x04 – 0x08	Driver status, returned as an unsigned 32-bit integer. 0 = No error 1 = Invalid parameter 2 = Parameter out of range 3 = Invalid pattern operation 4 = Not implemented 5 = Buffer too small for requested operation 6 = Internal error 7 = Undefined error 8 = Previous operation is still in progress.

Refer to the enum `bhi385_klio_param_driver_error_state_t` in `bhi385/bhi385_klio_param_defs.h` or the generated Doxygen for details regarding various error codes that may occur.

5.12 Additional code example notes

The struct `klio_runtime` contains runtime information required for the example.

```
typedef struct klio_runtime
{
    struct bhi385_dev *bhy2;
    bhi385_klio_sensor_state_t sensor_state;
    uint16_t max_patterns;
    uint16_t max_pattern_size;
    uint8_t ignore_insignificant_movement;
    uint8_t pattern_write_back_index;
    float* similarity_result_buf;
    uint8_t* similarity_idx_buf;
} klio_runtime_t;
```

The parameters `max_patterns` and `max_pattern_size` define the supported pattern capabilities of the BHI385 software. The parameter `ignore_insignificant_movement` attempts to prevent learning of very small movements.

The parameters `pattern_write_back_index`, `similarity_result_buf` and `similarity_idx_buf` are only used to demonstrate how to write back patterns and use the multiple similarity score feature. Regarding the buffers for the similarity score feature, if memory is limited in a real application, smaller buffers could be used with multiple calls to the similarity score function.

6 Using Klio with bhy2cli tool

A BHy2CLI tool is provided to quickly test and evaluated the Klio feature on BHI385 shuttle with the application board. Detailed setup and usage information can refer to the BHy2CLI User Guide document in the link: https://github.com/boschsensortec/BHy2CLI/tree/master/docs.

Two different interface types of BHy2CLI are provided: *i2c_bhy2cli.exe* and *spi_bhy2cli.exe*. Both are supported on the application board and customers can use based on their demand. Here using *spi_bhy2cli.exe* for an example.

To get the command help info of BHy2CLI tool:
> .\spi_bhy2cli.exe -h
or
> .\spi_bhy2cli.exe help

```
> .\spi_bhy2cli.exe -h
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Usage:
bhy2cli [<port>] [<port_name>] [<options>]
<port>: optional input parameter, trigger keyword
<port_name>: optional input parameter, use it when want to support running multiple applications in parallel
Options:
 -h OR help
      = Print this usage message
 version
      = Prints the version
 -v OR verb <verbose level>
      = Set the verbose level. 0 Error, 1 Warning, 2 Infos
 -b OR ramb <firmware path>
      = Reset, upload specified firmware to RAM and boot from RAM
        [equivalent to using "reset ram <firmware> boot r" successively]
 -n OR reset
      = Reset sensor hub
 -g OR boot <medium>
      = Boot from the specified <medium>: "r" for RAM
 -c OR actse <sensor id>:<frequency>[:<latency>][:<downsampling>]
      = Activate sensor <sensor id> at specified sample rate <frequency>,
       -latency <latency>, duration time <time>, sample counts <count>
       -At least <frequency> is a must input parameter
       -<latency> is optional
       -<downsampling> is optional, in case downsampling  = 0, it means there is no stream data is output
       -One or more sensors can be active by passing multiple actse options
       -id: sensor id
       -frequency(Hz): sensor ODR
       -latency(ms): sensor data outputs with a latency
       -downsampling(Hz): sensor downsampling value, it should be smaller than frequency to take effect
       -Eg:
       -actse 3:50::10
 ...
```

6.1 Basic bhy2cli commands

6.1.1 Trigger a soft reset to the sensor

> .\spi_bhy2cli.exe -n

```
> .\spi_bhy2cli.exe -n
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Reset successful
```

6.1.2 Reset, and then load the Klio firmware to RAM and boot

> .\spi_bhy2cli.exe -b <firmware>

```
> .\spi_bhy2cli.exe -b .\Bosch_Shuttle3_BHI385_bsxsam_lite_Klio_cyclic.fw
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Reset successful
Uploading 115972 bytes of firmware to RAM
Uploading firmware to RAM successful
Waiting for firmware verification to complete
Boot Status : 0x38: Host interface ready. Firmware verification done.
[D][META EVENT WAKE UP]; T: 0.366953125; Firmware initialized. Firmware version 5991
[D][META EVENT]; T: 0.366953125; Firmware initialized. Firmware version 5991
Booting from RAM successful
```

6.1.3 Show the list of virtual sensors

> .\spi_bhy2cli.exe -i

```
> .\spi_bhy2cli.exe -i
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Product ID     : 89
Kernel version : 2380
User version   : 8196
ROM version    : 5166
Power state    : sleeping
Host interface : SPI
Feature status : 0x4a
Boot Status : 0x38: Host interface ready. Firmware verification done.
Virtual sensor list.
Sensor ID |                      Sensor Name |  ID | Ver |  Min rate |  Max rate |
----------+-----------------------------------+-----+-----+-----------+-----------|
        1 |          Accelerometer passthrough | 205 |   1 |    1.5625 |  400.0000 |
```

4	Accelerometer corrected	241	1	1.5625 | 400.0000
5	Accelerometer offset	209	1	1.5625 | 1.5625
6	Accelerometer corrected wake up	192	1	1.5625 | 400.0000
10	Gyroscope passthrough	207	1	1.5625 | 400.0000
13	Gyroscope corrected	243	1	1.5625 | 400.0000
14	Gyroscope offset	208	1	1.5625 | 1.5625
15	Gyroscope wake up	194	1	1.5625 | 400.0000
37	Game rotation vector	252	1	1.5625 | 400.0000
38	Game rotation vector wake up	200	1	1.5625 | 400.0000
112	Klio	135	1	25.0000 | 50.0000
127	Klio log	136	1	1.5625 | 50.0000
136	Low Power Step counter	249	1	1.0000 | 1.0000
137	Low Power Step detector	248	1	1.0000 | 1.0000
143	Low Power Any motion wake up	191	1	1.0000 | 1.0000
153	Multi Tap Detector	182	1	1.5625 | 1.5625
154	Activity recognition for Wearables	237	1	1.5625 | 1.5625
156	Low Power Wrist Gesture wake up	228	1	1.0000 | 1.0000
158	Low Power Wrist Wear wake up	178	1	1.0000 | 1.0000
159	Low Power No Motion wake up	181	1	1.0000 | 1.0000

6.1.4 Enable a virtual sensor

> .\spi_bhy2cli.exe -c <Sensor ID>: <ODR in Hz>

```
> .\spi_bhy2cli.exe -c 4:25
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Sensor ID: 4, sample rate: 25.000000 Hz, latency: 0 ms


[D][META EVENT]; T: 207.839890625; Flush complete for sensor id 4
[D][META EVENT]; T: 207.840718750; Power mode changed for sensor id 4
[D][META EVENT]; T: 207.840718750; Sample rate changed for sensor id 4
[D][META EVENT]; T: 208.081593750; Accuracy for sensor id 4 changed to 0
[D]SID: 4; T: 208.161984375; x: 0.011719, y: -0.000732, z: 0.988770; acc: 0
[D]SID: 4; T: 208.202187500; x: 0.012207, y: -0.000488, z: 0.988525; acc: 0
[D]SID: 4; T: 208.242390625; x: 0.012451, y: -0.001465, z: 0.988037; acc: 0
[D]SID: 4; T: 208.282593750; x: 0.011719, y: -0.001465, z: 0.988281; acc: 0
[D]SID: 4; T: 208.322781250; x: 0.011719, y: -0.002197, z: 0.988770; acc: 0
[D]SID: 4; T: 208.362984375; x: 0.012207, y: -0.000244, z: 0.989258; acc: 0
```

The above command enables Accelerometer virtual sensor with ODR in 25Hz, other virtual sensors can also be enabled in the same way.

6.1.5 Enable multiple virtual sensors simultaneously

> .\spi_bhy2cli.exe -c <Sensor ID>: <ODR in Hz> -c <...>:<...>

```
> .\spi_bhy2cli.exe -c 4:25 -c 13:25
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Sensor ID: 4, sample rate: 25.000000 Hz, latency: 0 ms


Sensor ID: 13, sample rate: 25.000000 Hz, latency: 0 ms


[D][META EVENT]; T: 955.398703125; Flush complete for sensor id 4
[D][META EVENT]; T: 955.398703125; Flush complete for sensor id 13
[D][META EVENT]; T: 958.929562500; Power mode changed for sensor id 4
[D][META EVENT]; T: 958.929562500; Sample rate changed for sensor id 4
[D][META EVENT]; T: 958.952031250; Power mode changed for sensor id 13
[D][META EVENT]; T: 958.952031250; Sample rate changed for sensor id 13
[D][META EVENT]; T: 958.951328125; Flush complete for sensor id 4
[D][META EVENT]; T: 958.951328125; Flush complete for sensor id 13
[D][META EVENT]; T: 959.172968750; Accuracy for sensor id 4 changed to 1
[D][META EVENT]; T: 959.172968750; Accuracy for sensor id 13 changed to 1
[D]SID: 4; T: 959.213328125; x: 0.010986, y: -0.004639, z: 0.997070; acc: 1
```

```
[D]SID: 13; T: 959.213328125; x: 0.000000, y: 0.000000, z: 0.061035; acc: 1
[D]SID: 4; T: 959.253703125; x: 0.012451, y: -0.003662, z: 0.996582; acc: 1
[D]SID: 13; T: 959.253703125; x: 0.183105, y: 0.061035, z: 0.000000; acc: 1
[D]SID: 4; T: 959.294062500; x: 0.013428, y: -0.004639, z: 0.998535; acc: 1
[D]SID: 13; T: 959.294062500; x: -0.366211, y: -0.061035, z: 0.061035; acc: 1
[D]SID: 4; T: 959.334437500; x: 0.013916, y: -0.004639, z: 0.997559; acc: 1
[D]SID: 13; T: 959.334437500; x: 0.000000, y: -0.061035, z: 0.000000; acc: 1
```

The above command enables both Accelerometer and Gyroscope virtual sensors with ODR in 25Hz.

6.2 Klio related bhy2cli commands

The commands below configure and enable/disable the various aspects of the Klio sensor.

Table 16: Klio Commands Overview

Feature Class	Command	Description
Klio	**kstatus**	Get Klio driver status
	ksetstate	Set state of Klio
	kgetstate	Get the current state of Klio
	kreset	Reset all Klio state
	kldpatt	Load a pattern/adaptive pattern for recognition
	kenpatt	Enable pattern ids for recognition
	kdispatt	Disable pattern ids for recognition
	kdisapatt	Disable pattern adaption for given pattern ids
	kswpatt	Switch the pattern between the left and right hand
	kautldpatt	Automatically use learnt patterns for recognition
	ksetparam	Write Klio parameters
	kgetparam	Read Klio parameters
	ksetpattparam	Set Klio pattern parameters
	kgetpattparam	Get Klio pattern parameters
	ksimscore	Print similarity score for two patterns
	kmsimscore	Print similarity score for one or more stored patterns with reference to a reference pattern

Note: For the commands ***kenpatt, kdispatt, kswpatt, kdisapatt***, it is possible to pass multiple <pattern_id>'s as input by passing them as a comma separated list.
Refer to the flowchart below for the ***Klio*** command flow.

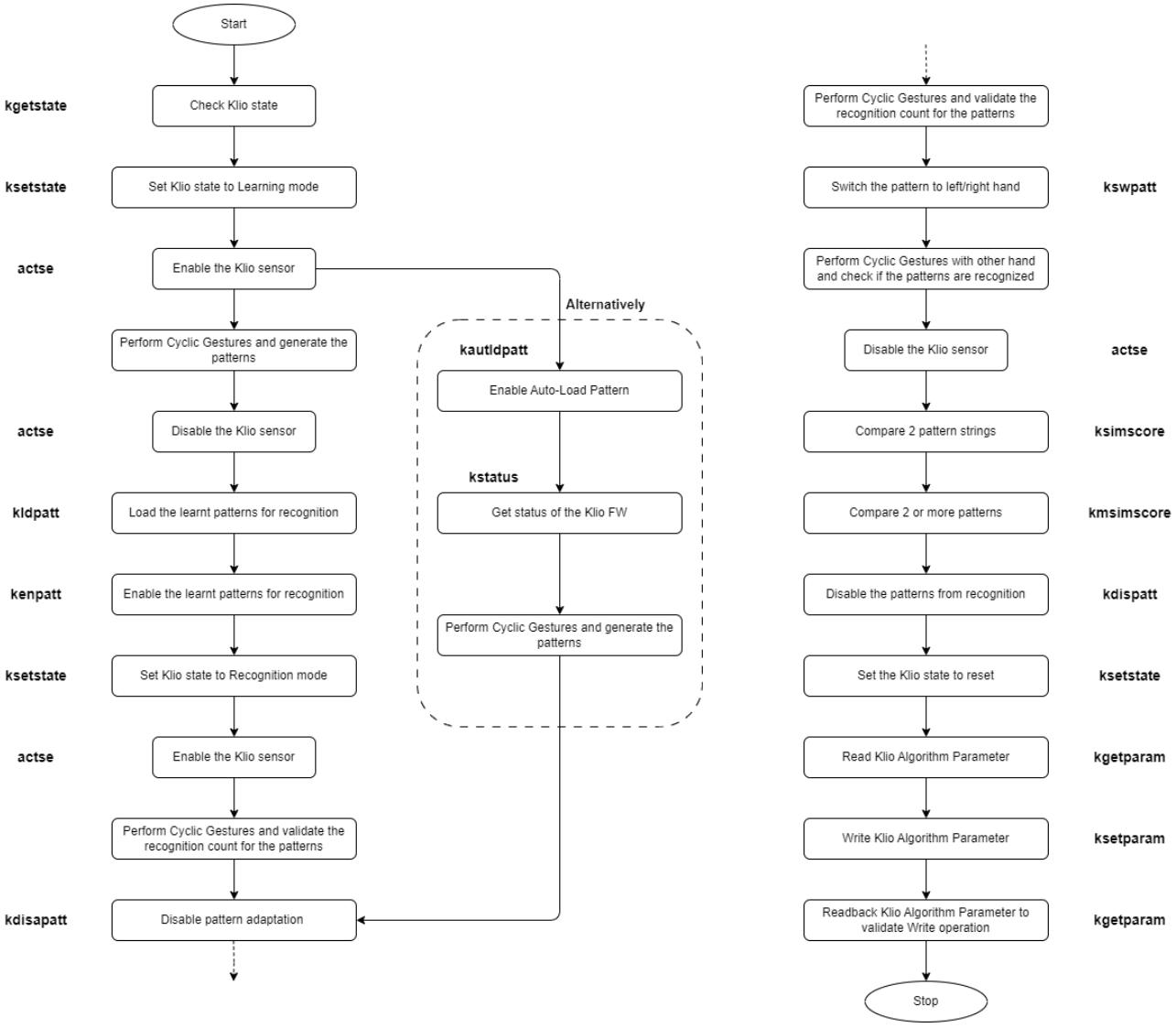


Figure 5: Klio Command Flow

6.2.1 Get and reset current Klio driver status

> .\spi_bhy2cli.exe kstatus

```
> .\spi_bhy2cli.exe kstatus
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Status: 0
```

The above command gets a status value of 0, indicating current Klio driver status is ok. Otherwise, non-zero value indicates Klio driver status is wrong.

6.2.2 Set Klio state

To prepare Klio for use, it is necessary to set it to a particular state. This can be accomplished using the "ksetstate" option, which is further described in the help message.

> .\spi_bhy2cli.exe ksetstate <le> <lr> <re> <rr>

```
> .\spi_bhy2cli.exe ksetstate 1 1 1 1
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Learning enabled    : 1
Learning reset      : 1
Recognition enabled : 1
Recognition reset   : 1
```

The above command enables both learning and recognition of Klio and resets the learning and recognition state.

```
> .\spi_bhy2cli.exe ksetstate 1 1 0 0
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Learning enabled    : 1
Learning reset      : 1
Recognition enabled : 0
Recognition reset   : 0
```

The above command only enables learning of Klio and resets the learning state.

```
> .\spi_bhy2cli.exe ksetstate 0 0 1 1
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
```

```
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Learning enabled    : 0
Learning reset      : 0
Recognition enabled : 1
Recognition reset   : 1
```

The above command only enables recognition of Klio and resets the recognition state.

6.2.3 Get current Klio state

To get the state of Klio, use the “kgetstate” option.

> .\spi_bhy2cli.exe kgetstate

```
> .\spi_bhy2cli.exe kgetstate
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Learning enabled    : 1
Learning reset      : 0
Recognition enabled : 1
Recognition reset   : 0
```

The above command shows that learning and recognition are both enabled.

6.2.4 Load a pattern for recognition

> .\spi_bhy2cli.exe kldpatt <index> <pattern>

```
> .\spi_bhy2cli.exe kldpatt 0
524231060334dca2400ad7233c456d4740abc3363f3cbce6bf1bd6b9bd914b06bfb652183dd8d0153f1ac94c3e91
eea23f242541bd3a340dbfcbf9ed3cbfcf0a41d70cadbec4b9994043dd3a40066f5abd29e0113ee80948beda24453
f2a080c3e98492b3a02ffb0be6194db3bee852dbe8ec13040ca52a23e74e2db3ea07b81bf6dfdd23ce7699f3ca32c
a7bedd04393e4ab3443e9ba20fbd6d484b3b
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
```

The above command loads a learnt pattern at index 0.

6.2.5 Enable pattern for recognition

> .\spi_bhy2cli.exe kenpatt <patterns>

```
> .\spi_bhy2cli.exe kenpatt 0 -c 112:25
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
```

```
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Enabled pattern 0 for recognition
Sensor ID: 112, sample rate: 25.000000 Hz, latency: 0 ms


[D][META EVENT]; T: 326.759281250; Flush complete for sensor id 112
[D]SID: 112; T: 333.590156250; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:0.619919 Score:0.233658]
[D]SID: 112; T: 333.832406250; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:0.932828 Score:0.849025]
[D]SID: 112; T: 334.115062500; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:1.133963 Score:0.870964]
[D]SID: 112; T: 334.316937500; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:1.298193 Score:0.879971]
[D]SID: 112; T: 334.478437500; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:1.441578 Score:0.886832]
[D]SID: 112; T: 334.639953125; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:1.554429 Score:0.884035]
[D]SID: 112; T: 334.801468750; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:1.669224 Score:0.891036]
[D]SID: 112; T: 334.882218750; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:1.769621 Score:0.899846]
[D]SID: 112; T: 334.962984375; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:1.883719 Score:0.918981]
[D]SID: 112; T: 335.084109375; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:2.002242 Score:0.924250]
[D]SID: 112; T: 335.205250000; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:2.103381 Score:0.921830]
```

The above command enables pattern at index 0. You can also enable multiple patterns simultaneously with a pattern index list separated with comma. For example: .\spi_*bhy2cli.exe kenpatt 0,1 -c 112:25*.

6.2.6 Disable pattern for recognition

> .\spi_bhy2cli.exe kdispatt <patterns>

```
> .\spi_bhy2cli.exe kdispatt 0
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Disabled pattern 0 from recognition
```

The above command disables pattern at index 0. You can also disable multiple patterns simultaneously with a pattern index list separated with comma. For example: *.\spi_bhy2cli.exe kdispatt 0,1*.

6.2.7 Automatically use learnt patterns for recognition

> .\spi_bhy2cli.exe kautldpatt <enable> <index>

```
> .\spi_bhy2cli.exe ksetstate 1 1 1 1  kautldpatt 1 0 -c 112:25
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Learning enabled    : 1
Learning reset      : 1
Recognition enabled : 1
Recognition reset   : 1


Klio auto load pattern Enabled, starting from index 0


Sensor ID: 112, sample rate: 25.000000 Hz, latency: 0 ms


[D][META EVENT]; T: 29.874015625; Flush complete for sensor id 112
[D]SID: 112; T: 35.308093750; Learning [Id:-1 Progress:0 Change:1]; Recognition[Id:255 Count:0.000000 Score:0.000000]
[D]SID: 112; T: 35.348453125; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:255 Count:0.000000 Score:0.000000]
[D]SID: 112; T: 35.631015625; Learning [Id:-1 Progress:20 Change:0]; Recognition[Id:255 Count:0.000000 Score:0.000000]
[D]SID: 112; T: 36.276875000; Learning [Id:-1 Progress:40 Change:0]; Recognition[Id:255 Count:0.000000 Score:0.000000]
[D]SID: 112; T: 36.397968750; Learning [Id:-1 Progress:20 Change:0]; Recognition[Id:255 Count:0.000000 Score:0.000000]
[D]SID: 112; T: 37.205312500; Learning [Id:-1 Progress:40 Change:0]; Recognition[Id:255 Count:0.000000 Score:0.000000]
[D]SID: 112; T: 38.012734375; Learning [Id:-1 Progress:60 Change:0]; Recognition[Id:255 Count:0.000000 Score:0.000000]
```

```
[D]SID: 112; T: 38.820109375; Learning [Id:-1 Progress:80 Change:0]; Recognition[Id:255 Count:0.000000 Score:0.000000]
[D]SID: 112; T: 39.627390625; Learning [Id:2 Progress:100 Change:0]; Recognition[Id:255 Count:0.000000 Score:0.000000]
[D]SID: 112; T: 39.627390625; Pattern learnt: 5242310603fdad80400ad7233cefd220406e46ab3ecfdfa8bf583bf9ba34c3f6bdf6f5113cd5b9f43ff2164abec65b1f3f7ed9ff3d24881ebe87d0f73b8e331241d7342bbf1151074082714a3f6209ed3e8f24f13c090eddba6854233eee338b3de1b29fbc671357bd6f12653a95cbadbd84918a3fc809b03ea1f9883e69176dbe400e6b3bd49eebbb215004be84212cbdf107a23c3f48f93c4e35d139
[D]SID: 112; T: 40.999859375; Learning [Id:-1 Progress:100 Change:0]; Recognition[Id:0 Count:0.602791 Score:0.187064]
[D]SID: 112; T: 41.363156250; Learning [Id:-1 Progress:100 Change:0]; Recognition[Id:0 Count:0.912584 Score:0.514584]
[D]SID: 112; T: 41.686062500; Learning [Id:-1 Progress:100 Change:0]; Recognition[Id:0 Count:1.125617 Score:0.739838]
[D]SID: 112; T: 41.928234375; Learning [Id:-1 Progress:100 Change:0]; Recognition[Id:0 Count:1.281731 Score:0.906680]
[D]SID: 112; T: 42.130062500; Learning [Id:-1 Progress:100 Change:0]; Recognition[Id:0 Count:1.423745 Score:0.952601]
[D]SID: 112; T: 42.251156250; Learning [Id:-1 Progress:100 Change:0]; Recognition[Id:0 Count:1.526557 Score:0.965035]
[D]SID: 112; T: 42.412625000; Learning [Id:-1 Progress:100 Change:0]; Recognition[Id:0 Count:1.655746 Score:0.967744]
[D]SID: 112; T: 42.533734375; Learning [Id:-1 Progress:100 Change:0]; Recognition[Id:0 Count:1.763177 Score:0.970638]
[D]SID: 112; T: 42.654828125; Learning [Id:-1 Progress:100 Change:0]; Recognition[Id:0 Count:1.870469 Score:0.969690]
[D]SID: 112; T: 42.816281250; Learning [Id:-1 Progress:100 Change:0]; Recognition[Id:0 Count:1.993995 Score:0.973125]
[D]SID: 112; T: 42.977734375; Learning [Id:-1 Progress:100 Change:0]; Recognition[Id:0 Count:2.109720 Score:0.971400]
```

The above command enables learning and recognition of Klio, as well as the automatic loading of patterns for the recognition feature. Additionally, it activates the Klio virtual sensor with an ODR of 25HZ. As evident from the printed messages, once the pattern is learned, it is automatically recognized correctly.

6.2.8 Switch pattern between left/right hand

> .\spi_bhy2cli.exe kswpatt <patterns>

```
> .\spi_bhy2cli.exe kswpatt 0 -c 112:25
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C


Switched pattern to opposite hand for pattern 0

Sensor ID: 112, sample rate: 25.000000 Hz, latency: 0 ms


[D][META EVENT]; T: 97.300859375; Flush complete for sensor id 112
[D]SID: 112; T: 99.836203125; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:0.632163 Score:0.290779]
[D]SID: 112; T: 100.240390625; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:0.938534 Score:0.746971]
[D]SID: 112; T: 100.563718750; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:1.162519 Score:0.882941]
[D]SID: 112; T: 100.846609375; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:1.331328 Score:0.885376]
[D]SID: 112; T: 101.048656250; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:1.481058 Score:0.882534]
[D]SID: 112; T: 101.169890625; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:1.606992 Score:0.891765]
[D]SID: 112; T: 101.291125000; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:1.748407 Score:0.892322]
[D]SID: 112; T: 101.412359375; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:1.860038 Score:0.894924]
[D]SID: 112; T: 101.574031250; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:2.002853 Score:0.890557]
[D]SID: 112; T: 101.695265625; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:2.116415 Score:0.891238]
[D]SID: 112; T: 101.856937500; Learning [Id:-1 Progress:0 Change:0]; Recognition[Id:0 Count:2.219653 Score:0.891663]
```

The above command switches pattern from left hand to right hand. By wearing the sensor on the right hand and performing the same action, Klio can accurately identify the pattern. You can also switch multiple patterns simultaneously with a pattern index list separated with comma. For example: *.\spi_bhy2cli.exe kswpatt 0,1*.

6.2.9 Get Klio parameter

> .\spi_bhy2cli.exe kgetparam <parameter id>

```
> .\spi_bhy2cli.exe kgetparam 8
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
```

```
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Parameter 8: 25
```

The above command gets the maximum number of patterns whose value is 25. For more information about parameter ids, please refer to `bhi385/bhi385_klio_param_defs.h`.

6.2.10 Set Klio parameter

> .\spi_bhy2cli.exe ksetparam <parameter id> <value>

```
> .\spi_bhy2cli.exe ksetparam 10 1
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Set value 1 for parameter id 10
```

The above command sets the learning process to disregard insignificant movements. For more information about parameter ids, please refer to `bhi385/bhi385_klio_param_defs.h`.

6.2.11 Get Klio pattern parameter

> .\spi_bhy2cli.exe kgetpattparam <index> <parameter id>

```
> .\spi_bhy2cli.exe kgetpattparam 0 0
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Pattern 0 Parameter 0 has value: 1.000000
```

The above command gets the exponent scaling factor for pattern 0, whose value is 1.0. If no pattern is loaded, the value is 0.0.
For more information about parameter ids, please refer to `bhi385/bhi385_klio_param_defs.h`.

6.2.12 Set Klio pattern parameter

> .\spi_bhy2cli.exe ksetpattparam <index> <parameter id> <value>

```
> .\spi_bhy2cli.exe ksetpattparam 0 0 1.5
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Pattern 0 parameter 0 set to value 1.5
```

The above command sets the exponent scaling factor for pattern at index 0 to 1.5, thereby making it more likely to be recognized.
For more information about parameter ids, please refer to `bhi385/bhi385_klio_param_defs.h`.

6.2.13 Get similarity score for two patterns

> .\spi_bhy2cli.exe ksimscore <pattern1> <pattern2>
Both pattern1 and pattern2 are bare hex byte string.

```
> .\spi_bhy2cli.exe ksimscore
5242310603fdad80400ad7233c74d92a405ba7233e689ba93d502585be5dda6bbed92ec73c4ac55abf48ce2cc06
26794bf7f080e3f46f18bbd3bdae93c41d01341dbf1e6bd3c2197bd5bc2143f842fbbbe5dd5493d4334953c67099c
bd7a3ce33dbabd17be4397b4bc3769de3a1978a3bd1da452bd4229fdbdd8be94bdae7626bed133fa3ac2af193d
b7d3093f860780bf9c640d3dec74433e7343da3a
5242310603fdad80400ad7233c76fec03fe41f823e996d5d3f72c5d8bd4c2438be4aca6f3c42f8c5bec8d152c08781
55c0b1e332be2d28ac3d41594c3dec191541fe9d77be4c99aabe7d8e2ebb6195a33d51dd603da7a2083de8a51ab
f881aed3e943d7fbd9843443e9d34073c4481cc3d74d415bea6a2df3b3ed496bc9870cabd7c88f73aa85e263da77
8413f7da673bf4e6d6e3da40d0c3d34044d3b
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Similarity: 0.971872
```

The above command compares two patterns represent in bare hex byte string. As evident from the printed messages, the similarity score between the two patterns is 0.971872, which indicates a relatively high degree of similarity. They are two variations of a left-hand dumbbell exercise – one with rotation of the wrist, and one with the palm facing forward.

6.2.14 Get similarity score for one or more stored patterns

> .\spi_bhy2cli.exe kmsimscore <base index> <comparison indices>

```
> .\spi_bhy2cli.exe kmsimscore 0 1,2
Host Interface : SPI
Copyright (c) 2025 Bosch Sensortec GmbH
Version 2.1.0 Build date: Nov  5 2025
Include sensor API with chip id: 0x7A
Include sensor API with chip id: 0x7C
Device found. Chip ID read 0x7C
Using pattern id 0 as reference: 1:0.059932 2:0.772561
```

The command above compares the pattern at index 0 with those at index 1 and 2. Ensure that you load the patterns using the "kldpatt" option before executing this command. As evident from the printed messages, there is a high similarity score between pattern 0 and pattern 2, indicating that they are similar. The similarity score varies between -1 and 1, with a higher score indicating greater similarity.

7 References

1. BHI385 Datasheet (BST-BHI385-DS000-04)
2. BHI360/BHI380/BHI385 SDK Quick Start Guide (BST-BHI3xx-AN000-07)
3. BHy2CLI Tool User Guide (BHy2CLI_User_Guide)
4. BHI385 Product webpage: https://www.bosch-sensortec.com/products/smart-sensor-systems/bhi385/

8 Legal disclaimer

i. Engineering samples

Engineering Samples are marked with an asterisk (*), (E) or (e). Samples may vary from the valid technical specifications of the product series contained in this data sheet. They are therefore not intended or fit for resale to third parties or for use in end products. Their sole purpose is internal client testing. The testing of an engineering sample may in no way replace the testing of a product series. Bosch Sensortec assumes no liability for the use of engineering samples. The Purchaser shall indemnify Bosch Sensortec from all claims arising from the use of engineering samples.

ii. Product use

Bosch Sensortec products are developed for the consumer goods industry. They may only be used within the parameters of this product data sheet. They are not fit for use in life-sustaining or safety-critical systems. Safety-critical systems are those for which a malfunction is expected to lead to bodily harm, death or severe property damage. In addition, they shall not be used directly or indirectly for military purposes (including but not limited to nuclear, chemical or biological proliferation of weapons or development of missile technology), nuclear power, deep sea or space applications (including but not limited to satellite technology).

The resale and/or use of Bosch Sensortec products are at the purchaser's own risk and his own responsibility. The examination of fitness for the intended use is the sole responsibility of the purchaser.

The purchaser shall indemnify Bosch Sensortec from all third party claims arising from any product use not covered by the parameters of this product data sheet or not approved by Bosch Sensortec and reimburse Bosch Sensortec for all costs in connection with such claims.

The purchaser accepts the responsibility to monitor the market for the purchased products, particularly with regard to product safety, and to inform Bosch Sensortec without delay of all safety-critical incidents.

iii. Application examples and hints

With respect to any examples or hints given herein, any typical values stated herein and/or any information regarding the application of the device, Bosch Sensortec hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights or copyrights of any third party. The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics. They are provided for illustrative purposes only and no evaluation regarding infringement of intellectual property rights or copyrights or regarding functionality, performance or error has been made.

9 Document history and modification

Rev. No	Chapter	Description of modification/changes	Date
1.0		Initial release	October 2026

Bosch Sensortec GmbH
Gerhard-Kindler-Straße 9
72770 Reutlingen / Germany

contact@bosch-sensortec.com
www.bosch-sensortec.com

Modifications reserved
Preliminary - specifications subject to change without notice
Document number: BST-BHI385-AN001-01