

APPLICATION NOTE



SCA11H HTTP API SAMPLE CODE

General Description

This document describes a simple method for interfacing with the SCA11H BCG sensor node through HTTP API. The method is described with a Python example code, which is included at the end of this document.

The example code can be used to read BCG info (`/bcg`) and calibration information (`/bcg/cali`), and it can as well be used to trigger empty and occupied bed calibrations.

The HTTP API itself is described in detail in HTTP API specification and it can be found on the Murata SCA11H product website. SSDP Discovery, for which a sample code is provided on the aforementioned website, can be used to find the node ip.

Output of the HTTP API sample code

The example code includes both BCG and calibration info HTTP GET functions. These can be substituted with different HTTP API requests in accordance to the HTTP API specification.

An example of a response to a HTTP GET to `/bcg` (BCG info):

```
{"version":"BCG Sensor_3.0.0.0", "mode":1, "pars":"7000,270,5000,0,1500,7", "dir":1}
```

version: BCG module version,

mode: BCG data (0), raw data (1), calibration phase 1 (2) and phase 2 (3).

pars: calibration parameters: `var_level1` (Signal High), `var_level2` (Signal Low), `stroke_vol` (Min Amplitude), `tentative_stroke_vol`, `signal_range` (Typ Amplitude), `to_micro_g` (Scale)

dir: measurement direction, 1 is normal, 0 is inverted

Examples of a response to a HTTP GET to `/bcg/cali` (calibration info):

```
{"status":-2} (no calibration done)
```

```
{"status":0, "phase":1, "step":255} (only empty bed calibration done)
```

status: calibration status info

phase: calibration phase

step: calibration step

The example code can also be used to trigger the desired embedded calibration process using `/bcg/cali` POST (message of type `{"phase": 2}`). As a response, an error number is returned.

Example return:

```
{"errno":0} (no errors, calibration successful)
```

sendMessage(...)

Sends a HTTP GET or PUSH command to the selected ip (node), depending on the inputs.

calibrationMenu(...)

A menu for choosing the desired function out of the following options:

1: Calibration info;

- returns the calibration parameters fetched from the node.

2: Empty bed calibration

- triggers phase 1 (empty bed) calibration in the node
- prints, whether calibration is successful or not

3: Occupied bed calibration

- triggers phase 2 (occupied bed) calibration in the node
- prints, whether calibration is successful or not

main(...)

Main function with an initial menu, where either (1) BCG info or (2) calibration menu can be requested. Prints out BCG info fetched from the BCG node if 1 is entered, runs calibrationMenu(...) if 2 is entered.

Example Python code

```

# THIS SOFTWARE IS PROVIDED BY MURATA "AS IS" AND
# ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
# LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
# FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
# MURATA BE LIABLE FOR ANY DIRECT, INDIRECT,
# INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
# HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
# STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
# IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
# POSSIBILITY OF SUCH DAMAGE.

# Python 3.x

import urllib.request
import urllib.error
import base64
import msvcrt      # Windows only!
import time

username = 'admin' # Username to access HTTP API
password = 'admin' # Password to access HTTP API

ip = input('Insert IP address: ') # IP of the BSN
ip = 'http://' + ip

# HTTP Basic authentication
headers = {
    'Authorization': 'Basic ' + base64.b64encode((username + ':' + password).encode('utf-8')).decode()
}

def sendMessage(path,body):
# Sends HTTP GET/POST to predefined IP
    # try:
    request = urllib.request.Request(ip+path, body, headers)
    result = urllib.request.urlopen(request)
    #print(result.code)
    #print(result.info())
    response = result.read().decode('utf-8')
    # except urllib.error.URLError:
    # response = 'Connection timed out.'
    return response

def calibrationMenu():
    print('CALIBRATION MENU\n\
    \'1\''\tCalibration info\n\
    \'2\''\tEmpty bed calibration\n\
    \'3\''\tOccupied bed calibration\n\
    \'esc\''\tExit calibration menu\n')
    while 1:
        keypress = msvcrt.getch()
        if keypress == chr(27).encode():
            break
        elif keypress == str('1').encode():
            # 4.12 Query BCG Calibration Status
            response = sendMessage('/bcg/cali',None)
            print(response)
        elif keypress == str('2').encode():
            # 4.13 Start BCG Calibration (phase 1 = empty bed)
            response = sendMessage('/bcg/cali',{'phase': 1}.encode('utf-8'))
            print(response)
            if response == '{"errno": 0}':
                print('Starting empty bed calibration. (60s)')
                print(str(0), end=" ", flush=True) # flush forces write
                for i in range(1,7):
                    time.sleep(10)
                    print(str(i*10), end=" ", flush=True )
                print('\nEmpty bed calibration finished')
            else:
                print('Empty bed calibration start failed')
                print(response) # Read the HTTP API for response definitions
        elif keypress == str('3').encode():
            # 4.13 Start BCG Calibration (phase 2 = occupied bed)
            response = sendMessage('/bcg/cali',{'phase': 2}.encode('utf-8'))
            print(response)
            if response == '{"errno": 0}':
                print('Starting occupied bed calibration. (60s)')
                print(str(0), end=" ", flush=True)
                for i in range(1,7):

```

```

        time.sleep(10)
        print(str(i*10), end=" ", flush=True )
        print('\nOccupied bed calibration finished')
    else:
        print('Occupied calibration start failed')
        print(response)      # Read the HTTP API for response definitions

def main():
    # Main Menu
    while 1:
        print('\nBCG HTTP API main menu\n\
\'1\'\tBCG info\n\
\'2\'\tCalibration menu\n\
\'esc\'\tQuit program\n')
        keypress = msvcrt.getch()
        if keypress == chr(27).encode():
            break
        elif keypress == str('1').encode():
            # 4.2 Query Basic BCG Info
            response = sendMessage('/bcg',None)
            print(response)      # Read the HTTP API for response definitions
        elif keypress == str('2').encode():
            calibrationMenu()

if __name__ == '__main__':
    main()

```

Rev.	Date	Change Description
1	09-June-17	First version.