SYNFIELD USER MANUAL

Version 0.9 – 08 April 2016

Produceret i projektet Økologi i sporet, som har fået tilskud fra Grønt Udviklings- og Demonstrations Program, GUDP under Fødevareministeriet.

Table of Contents

INTRODUCTION

SynField is a *cloud-based, sensor-logging ecosystem*. The ecosystem comprises of **SynField** devices (advanced data-loggers) that forward environmental data to the **SynField** web server.

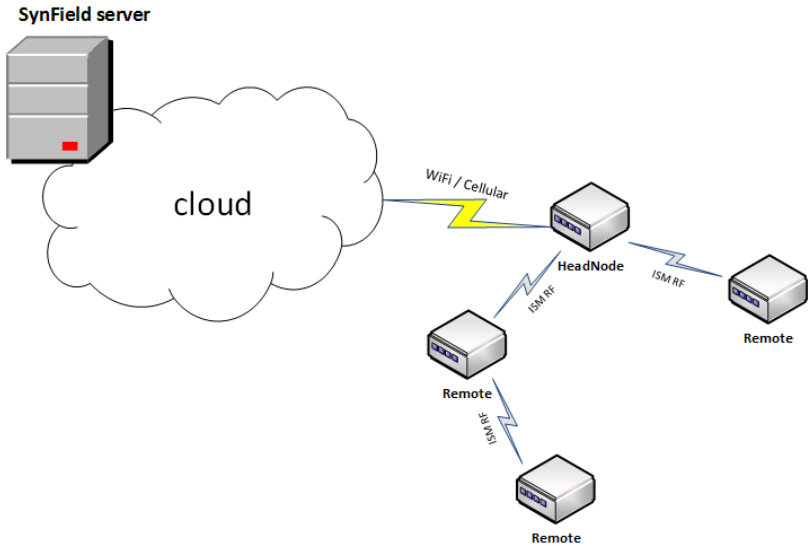

*Figure X. **SynField** ecosystem*

The **SynField** server portal enables the users to monitor the various environmental parameters and also provides the means to define sophisticated rules that can trigger alarms (i.e. SMS/email messages) or commands to be sent back to the remote device(s) to control specific actuators (i.e. solenoid valves, relay-switches).

Several **SynField** devices can be interconnected together by means of a multi-hop RF network (at the sub-GHz ISM band, 868/912 MHz) supporting Line-Of-Sight distances up to 700m.

FEATURES

- ✓ Sample, save and/or forward environmental parameters.
- ✓ Support plethora of sensors.
- ✓ Automatic remote control of actuators.
- ✓ Support several actuators.
- ✓ Internet connectivity through Wi-Fi or Cellular networks.
- ✓ Internode multi-hop RF connectivity (up to 700m LoS distances).
- ✓ Up to 20 remote nodes can be attached, utilizing the sub-GHz ISM RF channels.
- ✓ Solar/battery operated nodes.
- ✓ Advanced setup/control/monitoring application.

PRODUCT DESCRIPTION

There are two types of **SynField** devices: the Headnodes and the Remote devices (the device type is shown on the device label). At each **SynField** ecosystem, there is only one Headnode that establishes an internet connection to the **SynField** server and also acts as a gateway for the rest of the **SynField** devices (Remotes).

*Figure 1. View of **SynField** device (left) and two EC-5 soil moisture sensors in soil (right)*

INSTALLATION

Powering-up the device

The **SynField** devices are ready to operate upon arrival. The included battery is of Lithium Polymer type that remains charged for a long period of time.

In order to power up the device:

- Press the power-button.
- At start, the device enters the initialization phase (~10 seconds), during which the power-LED remains lit.
- After another 5 seconds, the power-LED flashes shortly and the device will:
    a. Initialize its network interface (WiFi/Cellular for Headnode devices, RF for the Remotes). Upon successful initialization, the power-LED flashes twice.
    b. Connect to the network (register to the WiFi/Cellular network for the Headnodes). Upon successful connection, the power-LED flashes twice.
    c. Send data to the **SynField** server. Upon successful data transmission, the power-LED flashes twice.

In order for the network connection to be established, the various network parameters on the Headnode should be set correctly (Wi-Fi SSID/password or Cellular APN settings). These settings can be configured using the SynControl application, as described in the “Configuration” section.

Installing sensors and actuators

The user can select several sensors and actuators, from a plethora of environmental sensors and actuators, to be connected to a particular **SynField** device. The supported sensors and actuators along with their characteristics are shown in Appendix A. In Appendix A, the various possible combinations of sensors/actuators per device is also shown.

Following the information found in Appendix A, the user should decide which sensors to connect and at which connector. The sensor cable should be inserted through the device box hole as shown in the following figure and then inserted to the appropriate connector.

Figure 3. Sensor installation

Depending on the sensor, it may be necessary to set some board jumpers accordingly. Appendix A, shows required jumper settings for each sensor, if any. The following figure depicts the diagram of the **SynField** Printed Circuit Board (PCB) that shows jumper positions.

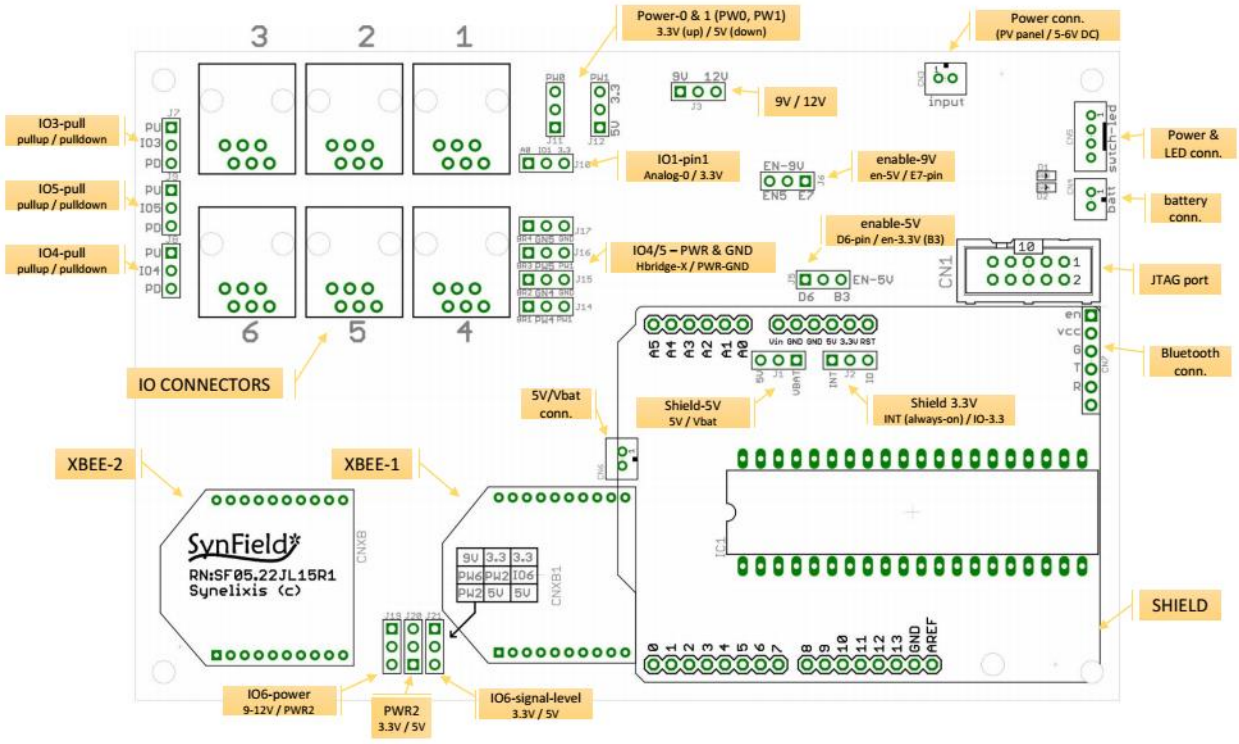

*Figure 4. **SynField** Printed Circuit Board (PCB) jumpers*

Lastly, the **SynField** system should be configured appropriately so as to read the sensor data. The "Configuration" section describes how to configure the device and also how to verify that the sensors are operating correctly.

CONFIGURATION

The **SynField** device requires some configuration parameters to be set in order to operate correctly. The parameters can be categorized as follows:

- Sensors/actuators parameters
- Networking parameters (Headnodes only)
- Attached remotes parameters (Headnodes only)

A detailed list of all configuration parameters can be found at Appendix B.

There are three ways to configure the **SynField** parameters:

1. Using the SynControl application.
2. Through the **SynField** portal (assuming that devices has succeeded to connect to the **SynField** server)
3. Using the on-board SD-card memory.

The third option is suggested only at situations that the other two options are not possible and is described at a different Application Note.

Configuration through the **SynField** portal is possible only for sensor/actuators parameters and is described at the "**SynField** Server Portal" section.

In this section we will describe the "**SynField** Control" (SynControl) application and how this can be used to configure the **SynField** device.

SynField Control (SynControl) application

Currently, the application is supported on devices running the Android operating system (4.03 onwards) and include a Bluetooth transceiver.

Firstly, you need to download and install the "**SynField** Control" application through the "Google Play Store".

When first starting the application, you will be asked to allow the application to turn on the Bluetooth (please allow) and then you'll get the start-up screen shown below.

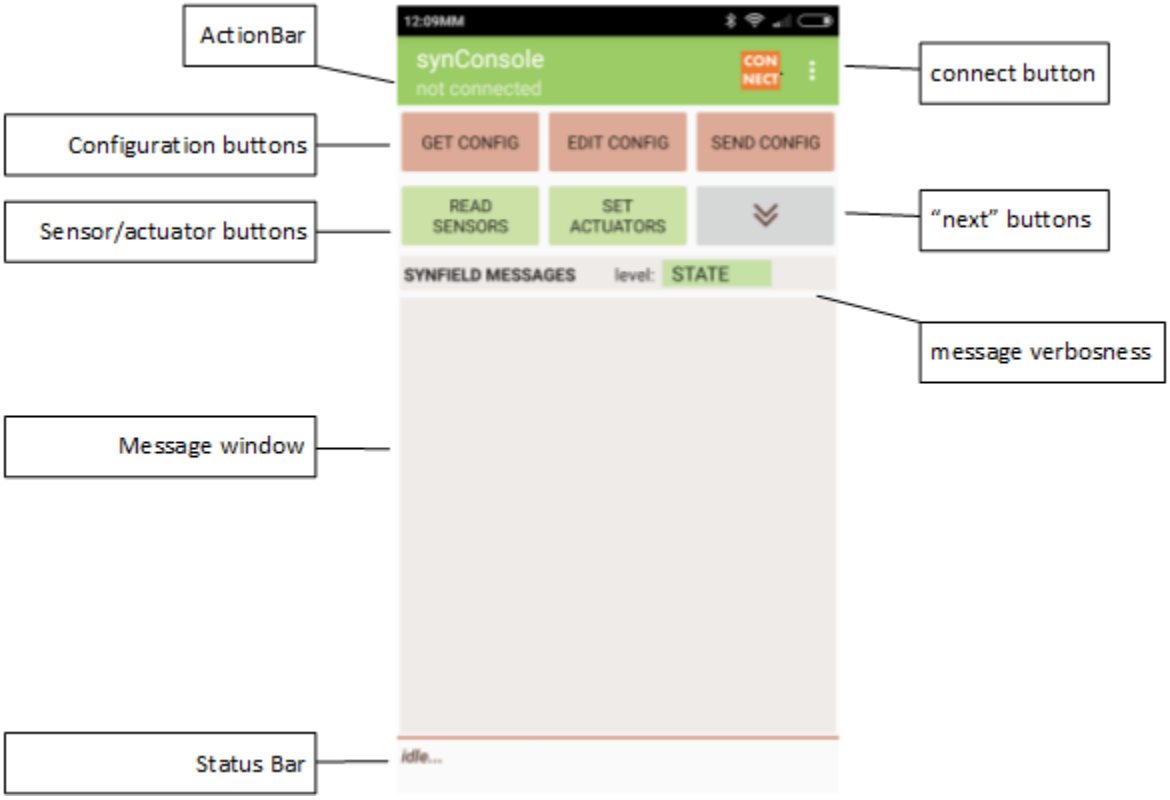

Figure 5. SynControl application: Start-up screen

By pressing, the "down-arrow" button, some extra buttons are revealed, as shown in the following figure. By pressing the "up-arrow" the original buttons are shown up again. The operation of each button is explained in the following subsections.

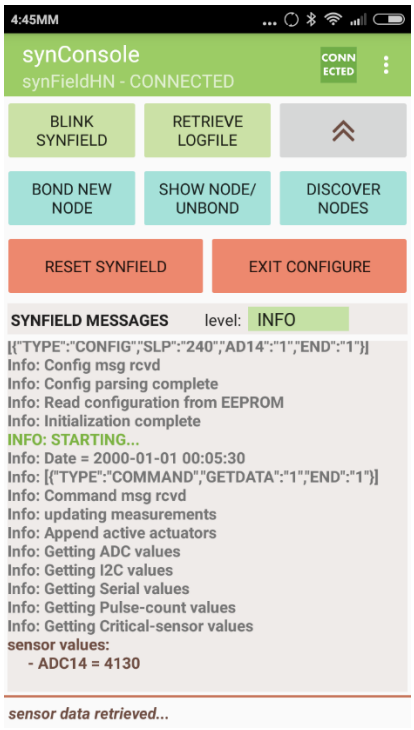

Figure 6. SynControl application: More buttons

Connecting to the SynField device

In order for the application to connect to the **SynField** device, do the following:

- Press, the orange “CONNECT” button on the top-right in order to connect to the **SynField** device.
- Turn on the **SynField** device. Upon power-up, the **SynField** device will turn on its Bluetooth transceiver for some time (typically ~30 seconds) and will try to connect to the SynControl application. If it cannot connect, it will power down its Bluetooth interface. So, make sure that if another **SynField** device is in range (10-20m), then it is either powered-down or it is powered up adequate time before.
- When the application locates the **SynField** device, it will automatically sent its pairing code. If, for some reason, the pairing key is not sent **automatically** and a dialog is showing that asks for a pairing code, then please type **“1234”** and press OK.
- Upon the establishment of the Bluetooth connection, the ActionBar (on the top of the application) will display the device type (**SynField**HN for Headnodes and **SynField**RT for Remotes).
- Following, the embedded software running on the **SynField** device, requires to establish connection with the application. This is accomplished with the exchange of messages over the Bluetooth connection and is performed at **5 to 10 seconds from the powering-up of the device**. If, for some reason (i.e. delayed Bluetooth connection) this did not occur, please repeat the operation.

Upon application connection, the status bar at the bottom and the ActionBar on the top, specifies that the device is connected.

Update configuration

After the application is connected, the user can update the device configuration as follows:

- Retrieve the current configuration of the device (press the "GET CONFIG" button).
- Edit the configuration parameters accordingly (press the "EDIT CONFIG" button).
- Send back the updated configuration to the device (press the "SEND CONFIG" button).

The application informs the user whether the operations are completed successfully. In case something goes wrong, an error message is displayed on the Status bar and on the Message window.

Monitor/control sensors and actuators

The user could verify that the attached sensors operate correctly by viewing the values that are read from the sensors. This is accomplished by pressing the "READ SENSORS" button (it is assumed that the application is connected to the Headnode) and wait for the values to be displayed on the Message window. Please note that the values displayed are the raw millivolt equivalent value read from the sensor. Appendix A shows respective millivolt equivalent ranges for each sensor. Modify sensed environmental parameters and verify that measured voltages are updated.

The user could also control the attached actuators (i.e. solenoid valves, switching relays) in order to verify that they are working properly. This is accomplished by just pressing by pressing the "SET ACTUATORS" button (it is assumed that the application is connected to the Headnode), a submenu with actuator ON/OFF buttons is displayed. Press the respective button and verify that the actuator changes to the expected state.

Setup Remote devices

As stated above, in the general case, the **SynField** ecosystem is comprised of a single Headnode device and several Remote nodes.

Upon setting up the system, the user must perform a one-time process that will connect the Remotes to the Headnode. This process is accomplished using the following procedure:

- "Bond"[1] each Remote to the Headnode.
    - Make sure that the Remote is in range and powered-off.
    - Press the "BOND NEW NODE" button (it is assumed that the application is connected to the Headnode)
    - Power-up the remote within 10 seconds from button pressing.
    - Upon "bonding" completion, a popup is displayed on the application, showing the local address obtained by the Remote (a number from 1 to 127).
- "Discover" bonded nodes. After "bonding" all Remotes, place the Remotes at their final positions and press the "DISCOVER NODES" button. Then a network training process is

[1] During the "Bonding" process, the RF module located on the Headnode searches for available **"un-bonded"** modules for ~10 seconds period. If an **"un-bonded"** Remote node is found, then a "bond" is formed, a local address (from 1 to 127) is assigned to the Remote and both RF modules are updated accordingly. Upon successful "bonding" the configuration on the Headnode is also updated.

performed, during which the network tree structure is formed and available routes from Headnode to Remotes are identified. Upon completion, a popup is displayed with the number of nodes that were “discovered”, which should be equal to the number of the remotes. If the number of discovered nodes is less than expected, then make sure that:

  - All the Remotes are powered-up and awake[2].
  - The range from one device to another is less of 700m if Line of Sight (LoS) is possible.
  - If LoS is not possible, then the RF range is reduced substantially and falls to about 100 meters depending on the terrain and environmental factors.

- If the position of the devices is altered considerably then it is advised to repeat the “discovery” process.

The user is also able to view the attached Remotes and “un-bond” some if required. Upon pressing the “SHOW NODE / UNBOND” button, a list of all attached Remotes is shown, where the user can “blink” or “un-bond” a specific or all Remotes as shown in the following figure.

Figure 7. Blink/UnBond Remote list

Note, that the Remote device should be in range and awake (see Footnote 2). Also**, note that the “UNBOND ALL” button performs local un-bonding only,** that is, only the Headnode RF module is updated of the un-bond process, thus the Remote believes that it is still bonded and cannot be re-bonded easily. For the time being, the solution is to send the respective RF module back in order to be re-programmed.

Various Functions

Through the SynControl application the user can also perform the following functions (it is assumed that the application is connected with the **SynField** device):

- Blink the device. By pressing the “BLINK **SYNFIELD**” button, the power-LED flushes 5 times.

[2] After powering-up, a Remote device performs initialization (~5 seconds) then goes into “sleep” for 5 seconds and then wakes-up in order to send data. It will remain at the “wake-up” state for 20 minutes, waiting for the Headnode to “talk” to it. If nothing is received during these 20 minutes, then the Remote will fall back to “sleep” for 20 minutes.

- Retrieve the log file. This is supported only on devices that include an on-board SD-card. By pressing the “RETRIEVE LOGFILE” button, the log-file that includes saved messages is retrieved and saved at the following location (it may differ slightly from device to device): /storage/sdcard0/Download/.
- Reset the device. Just press the “RESET **SYNFIELD**” button and the device will reboot in about 10 seconds.
- Exit configuration mode. By pressing the “EXIT CONFIGURE” button the device exits the configuration mode and returns to normal operation. However, note that the log messages from the device embedded software are still shown on the Message window, so that the user can observe device operation and possible points of failure (i.e. wrong network parameters). In order to re-enter into configuration mode, re-press the “ENTER CONFIGURE” button and restart the device.

SYNFIELD SERVER PORTAL

Various screens along with their description

Login

Management

Status/Control

Statistics/Plots

Rules

Disease prediction

APPENDIX A – Supported Sensors and Actuator

Sensor table

APPENDIX B – Configuration Parameters

Configuration Parameter table

SynField API: a brief description

1. Introduction

We provide you a set of REST web services in order to access your data or/and measurements. You could consume these services using any REST client or any other software.

HOST: api-synfield.synelixis.com

PORT: 80

The following table describes the provided SynField API operations:

HTTP METHOD	ENDPOINT	OPERATION
GET	/v1/sections/owner/{username}	Retrieve a collection of the sections that the user disposes. **Input**: the *username* is used as the key parameter. **Output**: the response body includes the coordinates of every section, the gateway, also called head node, (access) ID, a list of installed peripheral nodes (access) IDs, its unique ID, name and its type of crop. See section 2.1.
GET	/v1/gateway/{gatewayID}/sensors/	Retrieve a collection of the sensors that the gateway (head node) contains. **Input**: the *gatewayID*, which is provided in the first web service response body, is used as a key parameter. **Output**: the sensor ID, name and sensor status are provided for each sensor. Also, the relative service is offered. See section 2.2.
GET	/v1/gateway/{gatewayID}/measurements/	Retrieve a collection of the sensors' measurements that are installed in the gateway (head node). **Input**: the *gatewayID* parameter. This supports *paging* via query parameters parameters (*limit* & *offset*). **Output**: the response body contains as reference the next target request for paging issues. Any measurement contains its unique ID, the sensor ID, the type of service, its value and unit and its timestamp. See section 2.3.

EC–5
Soil Moisture Sensor

Operator's Manual

Decagon Devices, Inc.

Version: October 10, 2016 — 11:23:43

Decagon Devices, Inc.
2365 NE Hopkins Court
Pullman WA 99163

Phone: 509-332-5600
Fax: 509-332-5158
Website: www.decagon.com
Email: support@decagon.com or sales@decagon.com

Trademarks

Contents

1 Introduction

Thank you for choosing Decagon's EC-5 Volumetric Water Content sensor. This manual can help you understand the sensor features and ensure successful sensor operation. We hope you find the contents of this manual useful in understanding your instrument and maximizing its benefit to you.

There are several ways to contact Decagon if you ever need assistance with your product, have any questions, or feedback. Decagon has Customer Service Representatives available to speak with you Monday through Friday, between 7am and 5pm Pacific time.

Note: If you purchased your sensor through a distributor, please contact them for assistance.

Email:
support@decagon.com or **sales@decagon.com**

Phone:
509-332-5600

Fax:
509-332-5158

If contacting us by email or fax, please include as part of your message your instrument serial number, your name, address, phone, fax number, and a description of your problem or question.

Please read these instructions before operating your sensor to ensure that it performs to its full potential.

1.1 Warranty

The sensor has a 30-day satisfaction guarantee and a one-year warranty on parts and labor. Your warranty automatically validates upon receipt of the instrument.

1.2 Seller's Liability

Seller warrants new equipment of its own manufacture against defective workmanship and materials for a period of one year from the date of receipt of equipment.

Note: We do not consider the results of ordinary wear and tear, neglect, misuse, accident as defects.

The Seller's liability for defective parts shall in no event exceed the furnishing of replacement parts "freight on board" the factory where originally manufactured. Material and equipment covered hereby which is not manufactured by Seller shall be covered only by the warranty of its manufacturer. Seller shall not be liable to Buyer for loss, damage or injuries to persons (including death), or to property or things of whatsoever kind (including, but not without limitation, loss of anticipated profits), occasioned by or arising out of the installation, operation, use, misuse, nonuse, repair, or replacement of said material and equipment, or out of the use of any method or process for which the same may be employed. The use of this equipment constitutes Buyer's acceptance of the terms set forth in this warranty. There are no understandings, representations, or warranties of any kind, express, implied, statutory or otherwise (including, but without limitation, the implied warranties of merchantability and fitness for a particular purpose), not expressly set forth herein.

2 About EC-5

The EC-5 determines volumetric Water Content (VWC) by measuring the dielectric constant of the media using capacitance and frequency domain technology. The 70 MHz frequency minimizes salinity and textural effects, making this sensor accurate in almost any soil or soilless media. It arrives with factory calibration for mineral soils, potting soils, rockwool, and perlite included in this Operator's Manual.

The two-prong design and higher measurement frequency allows the EC-5 to measure VWC from 0 to 100% (VWC of saturated soils is generally 40 to 60% depending on the soil type) and allows accurate measurement of all soils and soilless medias with a wide range of salinities.

2.1 Specifications

Measurement Time: 10 ms (milliseconds)

Accuracy: at least 0.03 m^3/m^3 typical soils, up to 8 dS/m
With soil-specific calibration: $\pm.02$ m^3/m^3 ($\pm 2\%$)

Resolution: 0.001 m^3/m^3 VWC in mineral soils, 0.25% in growing media

Power Requirements: 2.5 VDC to 3.6 VDC @ 10 mA

Output: 10 to 40% of excitation voltage (250 to 1,000 mV at 2,500 mV excitation)

Operating Environment: -40 to 50 °C[1]

Range of Measurement: 0 to 100%

Sensor dimensions: 8.9 cm × 1.8 cm × 0.7 cm

Connector Types: 3.5 mm (stereo) plug or stripped & tinned lead wires (Pigtail)

[1]Sensors can be used at higher temperatures under certain conditions. Please contact Decagon for assistance.

Cable Length: 5 m standard; custom cable lengths up to 40 m available upon request

Data logger Compatibility (not exclusive):

- Decagon: Em50, Em50R, and Em50G
- Campbell Scientific: Any logger with serial I/O (CR10X, CR850, 1000, 3000, etc.)

3 Installing the Sensors

When selecting a site for installation, it is important to remember that the soil adjacent to the sensor surface has the strongest influence on the sensor reading and that the sensor measures the VWC.

Therefore any air gaps or excessive soil compaction around the sensor can profoundly influence the readings. Also, do not install the sensors adjacent to large metal objects such as metal poles or stakes. This can attenuate the sensors electromagnetic field and adversely affect output readings. Because the EC-5 has gaps between its prongs, it is also important to consider the size of the media you are inserting the sensor into. It is possible to get sticks, bark, roots or other material stuck between the sensor prongs, which will adversely affect readings. Finally, be careful when inserting the sensors into dense soil, as the prongs will break if excessive sideways force is used when pushing them in.

Installation Procedure

When installing the EC-5, it is best to maximize contact between the sensor and the soil.

- If you are installing sensors in a lightning prone area with a grounded data logger, please see our Application Note at www.decagon.com/lightning.
- Decagon advises that you test the sensors with your data logging device and software before going to the field.

The EC-5 sensor was designed for easy installation into the soil. After digging a hole to the desired depth, push the prongs on the sensor into undisturbed soil at the bottom of the hole or into the sidewall of the hole. Make sure that the prongs and black overmolding are buried completely. *Note: The sensor may be difficult to insert into extremely compact or dry soil. If you have difficulty inserting the sensor, try loosening the soil somewhat or wetting the soil. Never pound the sensor into the soil*

Carefully backfill the hole to match the bulk density of the surround-

ing soil. Be careful not to bend the black overmolding connecting the sensor to the cable.

To watch a video on proper installation of the sensor go to www.decagon.com/install.

Orientation

The sensor can be oriented in any direction. However, orienting the flat side perpendicular to the surface of the soil will minimize effects on downward water movement.

Removing the Sensor

When removing the sensor from the soil, do not pull it out of the soil by the cable! Doing so may break internal connections and make the sensor unusable.

3.1 Wiring

Figure 1: 3.5 mm Stereo Plug

The following software support the EC-5 sensor:

- ECH2O Utility 1.12 or greater
- ECH2O DataTrac 2.77 or greater

Connecting to a non-Decagon Logger

Customers may purchase EC-5 sensors for use with non-Decagon

data loggers. These sensors typically come configured with stripped and tinned (pigtail) lead wires for use with screw terminals. Refer to your distinct logger manual for details on wiring. Our Integrator's Guide gives detailed instructions on connecting the EC-5 sensor to non-Decagon loggers. Please visit www.decagon.com/support/literature for the complete Integrator's Guide.

Figure 2: Pigtail End Wiring

Pigtail End Wiring

Connect the wires to the data logger as Figure 3 shows, with the supply wire (white) connected to the excitation, the analog out wire (red) to a analog input, the bare ground wire to ground as illustrated in Figure 2.

Note: The acceptable range of excitation voltages is from 2.5 to 3.6 VDC.

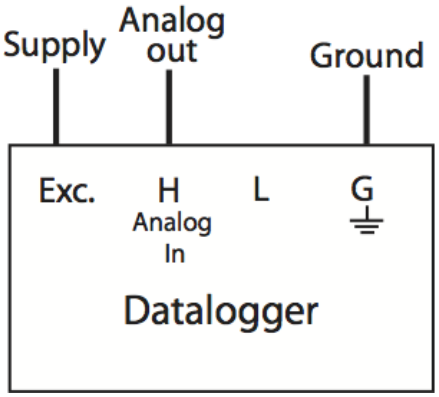

Figure 3: Connections

If your EC-5 is equipped with the standard 3.5 mm plug and you want to connect it to a non-Decagon data logger, you have two options. First, you can clip off the plug on the sensor cable, strip and tin the wires, and wire it directly into the data logger. This has the advantage of creating a direct connection with no chance of the sensor becoming unplugged; however, it cannot be easily used in the future with a Decagon readout unit or data logger.

The other option is to obtain an adapter cable from Decagon. The 3-wire sensor adapter cable has a connector for the sensor jack on one end, and three wires on the other end for connection to a data logger (this type of wire is often referred to as a "pigtail adapter"). Both the stripped and tinned adapter cable wires have the same termination as seen above; the white wire is excitation, red is output, and the bare wire is ground.

4 Connecting Sensors

Decagon designed the EC-5 sensor for use with our Em50 series data loggers, the Em5b data loggers, or the ProCheck handheld reader. The standard sensor (with a 3.5 mm "stereo plug" connector) quickly connects to and is easily configured within a Decagon logger or ProCheck.

The EC-5 sensor incorporates several features that also make it an excellent sensor for use with third party loggers. Customers may purchase the sensor with stripped and tinned wires (pigtail) for terminal connections.

The EC-5 sensor comes standard with a five meter cable. Customers may purchase sensors with custom cable lengths for an additional fee (on a per-meter fee basis). Obtaining custom length cables eliminates the need for splicing the cable (a possible failure point). The EC-5 is accurate with cable lengths up to 40 m.

Connecting to an Em50/Em50R Logger/Em50G/em5b

Decagon designed the EC-5 to work specifically with the Em50 data logger. Simply plug the 3.5 mm stereo plug connector directly into one of the five sensor ports. Next, configure the logger port for the EC-5 and set the measurement interval.

Connecting to ECH2O Utility

Please check your software version to ensure it will support the EC-5. To update your software to the latest version, please visit Decagon's software download site at www.decagon.com/support/downloads.

Note: You must use the ECH2O Utility, DataTrac 3 or a terminal program on your computer to download data from the logger to your computer.

5 Collecting Data

5.1 Data Logger Requirements

The EC-5 sensor is designed to work most efficiently with Decagon's 5-channel Em5b, Em50, or ProCheck handheld readout. All Decagon readout devices use a 3.0 V excitation.

The sensors however, may be adapted for use with other data loggers, such as those from Campbell Scientific, Inc., for example. The EC-5 requires an excitation voltage in the range of 2.5 to 3.6 V. The sensors produce an output voltage that depends on the dielectric constant of the medium surrounding the sensor, and ranges between 10 and 50% of the excitation voltage. Any data logger which can produce a 2.5 to 3.6 V excitation with approximately 10 millisecond duration and read a volt level signal with 12-bit or better resolution should be compatible with the EC-5 sensor. The current requirement for the EC-5 is 10 mA at 2.5 V.

We designed the EC-5 sensor for use with data loggers and readout devices that provide short excitation pulses, leaving the sensors turned off most of the time. Continuous excitation not only wastes battery power, but may, under certain circumstances, cause the sensor to exceed government specified limits on electromagnetic emissions. Do not continuously power the EC-5 sensor.

5.2 Sample Program

The following program is an example that can be used with the Campbell Scientific CR10X data logger and our EC-5 sensor at a 2,500 mV excitation:

```
;{CR10X}
; Example ECH2O Data Logger Program for CR10X

; Wiring:
```

```
; White: Excitation Channel 1
; Red: Input Single Ended Channel 1
; Black: Ground

*Table 1 Program
01: 1     Execution Interval (seconds)

; Factory calibration equations for ECH2O
; probes convert mV output of ECH2O to
; volumetric water content (VWC, m3/m3)

; EC-5:  VWC = 0.00119 * mV - 0.400

1:  Excite-Delay (SE) (P4)
 1: 1      Reps
 2: 5      2500 mV Slow Range
 3: 1      SE Channel
 4: 1      Excite all reps w/Exchan 1
 5: 1      Delay (0.01 sec units)
 6: 2500 mV      Excitation
 7: 1      Loc [ Probe_VWC ]
 8: .00119      Multiplier
 9: -.4      Offset

*Table 2 Program
02: 0.0000 Execution Interval (seconds)

*Table 3 Subroutines

End Program
```

5.3 SCWin (Short Cut) Directions

The following are instructions for using the Campbell Scientific SCWin (Short Cut) program to read the EC-5 soil moisture sensor.

1. Download EchoCSI.zip from http://www.decagon.com/appnotes/EchoCSIappnote.pdf.

2. Unzip the folder EchoCSI.zip.
3. Locate the file containing SCWin.exe. It should be in C:\Program Files\Campbellsci\SCWin. Place the following files from the unzipped EchoCSI.zip folder into the folder with SCWin.exe.
   - AM1632Z.MUX
   - AM416Z.MUX
   - EC10.SCS
   - EC101632.SCS
   - EC10416.SCS
   - EC20.SCS
   - EC201632.SCS
   - EC20416.SCS
   - EC5.SCS
   - EC5632.SCS
   - EC5416.SCS
   - SCWIN-DECAGON.CNT
   - SCWIN-DECAGON.HLP

   Note: If you are not able to find this directory path, search for the folder that contains SCWIN.exe and place the files into that folder.
4. Open up SCWin.exe (Short Cut). If you are using a V.3 copy of LoggerNet, there is a tab for SCWin (Short Cut) on the tool bar.
5. Select New to start a new program to read the EC-5.
   (a) Select the data logger you will be using to read the sensors.
   (b) Select the measurement interval (a shorter measurement interval, i.e. 1 sec., is sometimes desirable when testing the sensor).

6. Click on Sensors (this should open a new page with a file tree on it).
7. Under the "Sensors" file tree, double-click on "Meteorological" and then select "Soil Moisture."
8. Choose "EC-5" Sensor.

5.4 Removing the Sensor

When removing the sensor from the soil, do not pull it out of the soil by the cable. Doing so may break internal connections and make the sensor unusable.

6 Calibration

Decagon's ECH2O Utility and DataTrac3 automatically apply factory calibrations to the sensor output data. However, this general calibration may not be applicable for all soil types. For added accuracy we encourage our customers to perform soil-specific calibrations.

Which calibration equation you use depends on where you use it. If you use it with sensors connected to a non-Decagon data logger you will need to use the calibration appropriate to your excitation voltage. If you use any Decagon software (DataTrac3, ECH2O Utility, etc.) or the user calibration menu in the ProCheck, you must use the RAW calibration. The difference between the two is the slope constant. To increase the resolution of the sensor output, Decagon uses all available increments of the 12-bit number (value of 4096) where the measurement is stored. Thus, the output of the sensors read by the ProCheck and Decagon loggers must be multiplied by 0.61 and the 2,500 mV slope to give the right value.

6.1 Sensor Calibration Values

Following is a list of the both the millivolt and RAW calibration values for the EC-5, where ϵ is the VWC, mV is the millivolt output of the sensor, and where x is the RAW sensor output.

The EC-5 is much less sensitive to variation in texture and electrical conductivity because it runs at a much higher measurement frequency. Therefore, its general calibration equation should apply for all mineral soils up to 8 dS/m saturation extract. Its calibration equations are shown below for mineral soil, potting soil, and rockwool growing media.

Dielectric Permittivity

Dielectric permittivity can be used to determine VWC using external published equations such as the Topp equation. Dielectric

permittivity is given by:

$$\varepsilon = 1/(-1.10570 \text{ x } 10^{-9} * RAW^3 + 3.575 \text{ x } 10^{-6} * RAW^2 - 3.9557 \text{ x } 10^{-3} * RAW + 1.53153) \quad (1)$$

where RAW is the output from the Decagon data logger using 3V excitation. If you are using a non-Decagon data logger, dielectric permittivity is given by

$$\varepsilon = 1/(-3.3326 \text{ x } 10^{-9} * mV^3 + 7.0218 \text{ x } 10^{-6} * mV^2 - 5.11647 \text{ x } 10^{-3} * mV + 1.30746) \quad (2)$$

Mineral Soils

According to our tests, a single calibration equation generally suffices for all mineral soil types with electrical conductivities from 0.1 dS/m to 10 dS/m saturation extract. VWC (θ) is given by:

$$\theta = 8.5 * 10^{-4} * RAW - 0.48 \quad (3)$$

where RAW is the output from the Decagon data logger using 3 V excitation. If you are using a non-Decagon data logger, VWC is given by:

$$\theta = 11.9 * 10^{-4} * mV - 0.401 \quad (4)$$

where mV is the output of the sensor when excited at 2,500 mV. Please note that the equation reachs a maximum at ∼60% VWC in pure water. To display data on a scale from 0 to 100%, VWC should be modeled with a quadratic equation (which would result in a 100% VWC in water), but a linear equation fits the mineral soil VWC range as well as the quadratic, and linear equations are easier to deal with, especially since mineral soil typically saturates at ∼40 to 50% VWC.

Potting soil

The following equations can be used to convert EC-5 output to water content in potting soil. We tested several types of potting soil (Sunshine mix, Miracle Grow Potting Mix, and Custom Nursery soil) at several salinities and found that VWC is given by:

$$\theta = 1.3 \text{ x } 10^{-3} * RAW - 0.696 \quad (5)$$

for a Decagon data logger or

$$\theta = 2.11 \text{ x } 10^{-3} * mV - 0.675 \quad (6)$$

for a data logger with 2,500 mV excitation.

Rockwool

We calibrate the EC-5 in Grodan Master rockwool with solution electrical conductivities of 0.2, 1.0, 1.5, 2.0, and 4.5 dS/m. VWC can be calculated using

$$\theta = 6.28 \text{ x } 10^{-7} * RAW^2 + 1.37 \text{ x } 10^{-4} * RAW - 0.183 \quad (7)$$

for a Decagon data logger or

$$\theta = 2.63 \text{ x } 10^{-6} * mV^2 + 5.07 \text{ x } 10^{-4} * mV - 0.0394 \quad (8)$$

for a data logger with 2,500 mV excitation.

Note: These calibration constants only apply to 2,500 mV excitations; use of these numbers with any other excitation voltage results in erroneous readings!

7 Maintenance and Troubleshooting

If you encounter problems with the EC-5, they most likely manifest themselves in the form of incorrect or erroneous readings. Before contacting Decagon about the sensor, do the following:

- Check to make sure the connections to the data logger are both correct and secure.
- Ensure that your data logger batteries are not dead or weakened.

  If you encounter problems that are not due to the data logger, please contact Decagon at (509) 332-5600 or at support@decagon.com.

8 Declaration of Conformity

Application of Council Directive:	2004/108/EC and 2011/65/EU
Standards to which conformity is declared:	EN 61326-1:2013 and EN 50581:2012
Manufacturer's Name:	Decagon Devices, Inc. 2365 NE Hopkins Ct. Pullman, WA 99163 USA
Type of Equipment:	Soil Moisture Sensor
Model Number:	EC-5
Year of First Manufacture:	2001

This is to certify that the EC-5 ECH2O soil moisture sensor, manufactured by Decagon Devices, Inc., a corporation based in Pullman, Washington, USA meets or exceeds the standards for CE compliance as per the Council Directives noted above. All instruments are built at the factory at Decagon and pertinent testing documentation is freely available for verification.

Index

GET	/v1/gateway/{gatewayID}/measurements/{fromDate}/{untilDate}/	Retrieve a collection of the sensors' measurements that are installed in the gateway (head node) at a specific date range. **Input**: the *gatewayID*, *fromDate* and *untilDate* parameters. It supports *paging* via query parameters parameters (*limit* & *offset*). **Output**: The response body contains as reference the next target request for paging issues. Any measurement contains its unique ID, the sensor ID, the type of service, its value and unit and its timestamp. See section 2.4.
GET	/v1/peripheral/{peripheralID/sensors	Retrieve a collection of the sensors that the specific peripheral node contains. **Input**: the *peripheralID* parameter, which is included in the response body of the first web service. **Output**: the sensor ID, the name and the sensor status are provided for each sensor. Also, the relative service ID and name are provided. See section 2.5.
GET	/v1/peripheral/{peripheralID}/measurements	Retrieve a collection of the sensors' measurements that are installed in the peripheral node. **Input**: the *peripheralID* parameter. This supports *paging* via query parameters (*limit* & *offset*). **Output**: the response body contains as reference the next target request for paging issues. Any measurement contains its unique ID, the sensor ID, the type of service, its value and unit and its timestamp. See section 2.6.
GET	/v1/peripheral/{peripheralID}/measurements/{fromDate}/{untilDate}	Retrieve a collection of the sensors' measurements that are installed in the peripheral node at a specific date range. **Input**: the *peripheralID*, *fromDate* and *untilDate* parameters. It supports *paging* via query parameters (*limit* & *offset*). **Output**: The response body contains as reference the next target request for paging issues, if it needs. Any measurement contains its unique ID, the sensor ID, the type of service, its value and unit and its timestamp. See section 2.7.

2. API operations

2.1. Retrieve a collection of sections per user (owner)

Method: GET

Endpoint: http://api-synfield.synelixis.com/v1/sections/owner/guest

Query parameters: None

Headers:

- Accept: application/json
- Authorization: Basic {token}[1]

Response

Status Code: 200 OK


```
{
    "response": {
        "sections": [
            {
                "crop_type": "Olives",
                "elevation": 151.947,
                "headNode": "00:06:66:69:32:32",
                "id": 5,
                "latitude": 36.6146,
                "longitude": 23.0198,
                "name": "Foutia",
                "peripheralNodeList": [
                    "0013A20040A2873F"
                ],
                "timezone": "Europe/Athens",
                "user": 14
            }
        ]
    }
}
```


2.2. Retrieve a collection of installed sensors in the Head Node (gateway)

Method: GET

Endpoint: http://api-synfield.synelixis.com/v1/gateway/00:06:66:69:32:32/sensors/

Query parameters: None

Headers:

- Accept: application/json

[1] **token** is a string that is derived from the base64 conversation of the *{username}:{password}* , where username and password reflects the user credentials.

- Authorization: Basic {token}

Response

Status Code: 200 OK


```
{
  "response": {
    "sensors": [
      {
        "enabled": "True",
        "id": 339,
        "name": "Battery",
        "service": "Battery"
      },
      {
        "enabled": "True",
        "id": 340,
        "name": "LM61",
        "service": "System temperature"
      },
      {
        "enabled": "True",
        "id": 341,
        "name": "Solar panel",
        "service": "Solar radiation level"
      },
      {
        "enabled": "True",
        "id": 343,
        "name": "WindMeter",
        "service": "Wind speed"
      },
      {
        "enabled": "True",
        "id": 344,
        "name": "WindDirection-v2",
        "service": "Wind direction"
      },
      {
        "enabled": "True",
        "id": 345,
        "name": "RainMeter",
        "service": "Rain"
      },
      {
        "enabled": "True",
        "id": 346,
        "name": "SLHT5-Air",
        "service": "Air temperature"
      },
      {
        "enabled": "True",
        "id": 347,
        "name": "SLHT5-Air",
        "service": "Air humidity"
      },
      {
        "enabled": "True",
        "id": 348,
```

```
        "name": "LeafWetnessLib",
        "service": "Foil moisture"
      }
    ]
  }
}
```


2.3. Retrieve a collection of sensors' measurements in the Head Node (gateway)

Method: GET

Endpoint example:

http://api-synfield.synelixis.com/v1/gateway/00:06:66:69:32:32/measurements/?offset=0&limit=5

Query parameters:

- offset
  - *type*: integer
  - *default value (in 1st request)*: 0
  - *usage:* optional
- limit
  - *type*: integer
  - *default value*: 500
  - *usage:* optional

Headers:

- Accept: application/json
- Authorization: Basic {token}

Response

Status Code: 200 OK


```
{
    "response": {
        "links": [
            {
                "ref": "next",
                "url": "http://api-
synfield.synelixis.com/v1/gateway/00:06:66:69:32:32/measurements/?offset=5&limit=5"
            }
        ],
        "measurements": [
            {
                "id": 15792,
                "sensor_id": 339,
                "service": "Battery",
                "timestamp": "2015-03-06 16:32:10",
                "unit": "Volt",
                "value": 3.77
```

```
            },
            {
                "id": 15790,
                "sensor_id": 340,
                "service": "System temperature",
                "timestamp": "2015-03-06 16:32:10",
                "unit": "C",
                "value": 16.8
            },
            {
                "id": 15791,
                "sensor_id": 341,
                "service": "Solar radiation level",
                "timestamp": "2015-03-06 16:32:10",
                "unit": "Levels",
                "value": 1
            },
            {
                "id": 15795,
                "sensor_id": 339,
                "service": "Battery",
                "timestamp": "2015-03-06 16:34:47",
                "unit": "Volt",
                "value": 3.77
            },
            {
                "id": 15793,
                "sensor_id": 340,
                "service": "System temperature",
                "timestamp": "2015-03-06 16:34:47",
                "unit": "C",
                "value": 16.8
            }
        ],
        "name": "Foutia",
        "timezone": "Europe/Athens"
    }
}
```


2.4. Retrieve a collection of sensors' measurements in the Head Node (gateway) at a specific date range

Method: GET

Endpoint example:

http://api-synfield.synelixis.com/v1/gateway/00:06:66:69:32:32/measurements/2015-05-06/2015-05-06/?offset=0&limit=5

Query parameters:

- offset
  - *type*: integer
  - *default value (in 1st request)*: 0
  - *usage:* optional

- limit
    - *type*: integer
    - *default value*: 500
    - *usage:* optional

Headers:

- Accept: application/json
- Authorization: Basic {token}

Response

Status Code: 200 OK


```
{
    "response": {
        "links": [
            {
                "ref": "next",
                "url": "http://api-
synfield.synelixis.com/v1/gateway/00:06:66:69:32:32/measurements/2015-05-06/2015-05-
06/?offset=5&limit=5"
            }
        ],
        "measurements": [
            {
                "id": 43210,
                "service": "Battery",
                "timestamp": "2015-05-06 00:12:27",
                "unit": "Volt",
                "value": 4.12
            },
            {
                "id": 43208,
                "service": "System temperature",
                "timestamp": "2015-05-06 00:12:27",
                "unit": "C",
                "value": 18.5
            },
            {
                "id": 43209,
                "service": "Solar radiation level",
                "timestamp": "2015-05-06 00:12:27",
                "unit": "Levels",
                "value": 1
            },
            {
                "id": 43211,
                "service": "Wind speed",
                "timestamp": "2015-05-06 00:12:27",
                "unit": "Beaufort",
                "value": 1
            },
            {
                "id": 43206,
                "service": "Wind direction",
                "timestamp": "2015-05-06 00:12:27",
```

```
                "unit": "",
                "value": "Southeast"
            }
        ],
        "name": "Foutia",
        "timezone": "Europe/Athens"
    }
}
```


2.5. Retrieve a collection of installed sensors in any Peripheral Node

Method: GET

Endpoint example:

http://api-synfield.synelixis.com/v1/peripheral/0013A20040A2873F/sensors

Query parameters: None

Headers:

- Accept: application/json
- Authorization: Basic {token}

Response

Status Code: 200 OK


```
{
    "response": {
        "sensors": [
            {
                "enabled": "True",
                "id": 349,
                "name": "Battery",
                "service": "Battery",
                "service_id": 350
            },
            {
                "enabled": "True",
                "id": 350,
                "name": "LM61",
                "service": "System temperature",
                "service_id": 351
            },
            {
                "enabled": "True",
                "id": 351,
                "name": "Solar panel",
                "service": "Solar radiation level",
                "service_id": 352
            }
        ]
    }
}
```

2.6. Retrieve a collection of sensors' measurements in any Peripheral Node

Method: GET

Endpoint example:

http://api-synfield.synelixis.com/v1/peripheral/0013A20040A2873F/measurements?limit=4

Query parameters

- offset
  - *type*: integer
  - *default value (in 1st request)*: 0
  - *usage:* optional
- limit
  - *type*: integer
  - *default value*: 500
  - *usage:* optional

Headers:

- Accept: application/json
- Authorization: Basic {token}

Response

Status Code: 200 OK


```
{
    "response": {
        "links": [
            {
                "ref": "next",
                "url": "http://api-
synfield.synelixis.com/v1/peripheral/0013A20040A2873F/measurements?offset=4&limit=4"
            }
        ],
        "measurements": [
            {
                "id": 22335,
                "sensor_id": 349,
                "service": "Battery",
                "timestamp": "2015-04-08 14:45:42",
                "unit": "Volt",
                "value": 3.74
            },
            {
                "id": 22333,
                "sensor_id": 350,
                "service": "System temperature",
                "timestamp": "2015-04-08 14:45:42",
                "unit": "C",
                "value": 6.1
            },
```

```
            {
                "id": 22334,
                "sensor_id": 351,
                "service": "Solar radiation level",
                "timestamp": "2015-04-08 14:45:42",
                "unit": "Levels",
                "value": 3
            },
            {
                "id": 22345,
                "sensor_id": 349,
                "service": "Battery",
                "timestamp": "2015-04-08 15:07:02",
                "unit": "Volt",
                "value": 3.74
            }
        ],
        "name": "Foutia",
        "timezone": "Europe/Athens"
    }
}
```


2.7. Retrieve a collection of sensors' measurements in the Peripheral Node at a specific date range

Method: GET

Endpoint example:

http://api-synfield.synelixis.com/v1/peripheral/0013A20040A2873F/measurements/2015-05-02/2015-05-03?offset=132&limit=4

Query parameters

- offset
  - *type*: integer
  - *default value (in 1st request)*: 0
  - *usage:* optional
- limit
  - *type*: integer
  - *default value*: 500
  - *usage:* optional

Headers:

- Accept: application/json
- Authorization: Basic {token}

Response

Status Code: 200 OK

```
{
    "response": {
        "links": [
            {
                "ref": "next",
                "url": "http://api-
synfield.synelixis.com/v1/peripheral/0013A20040A2873F/measurements/2015-05-02/2015-05-
03?offset=136&limit=4"
            }
        ],
        "measurements": [
            {
                "id": 40643,
                "sensor_id": 349,
                "service": "Battery",
                "timestamp": "2015-05-02 17:57:10",
                "unit": "Volt",
                "value": 4.14
            },
            {
                "id": 40641,
                "sensor_id": 350,
                "service": "System temperature",
                "timestamp": "2015-05-02 17:57:10",
                "unit": "C",
                "value": 19.5
            },
            {
                "id": 40642,
                "sensor_id": 351,
                "service": "Solar radiation level",
                "timestamp": "2015-05-02 17:57:10",
                "unit": "Levels",
                "value": 5
            },
            {
                "id": 40655,
                "sensor_id": 349,
                "service": "Battery",
                "timestamp": "2015-05-02 18:18:05",
                "unit": "Volt",
                "value": 4.14
            }
        ],
        "name": "Foutia",
        "timezone": "Europe/Athens"
    }
}
```