Integration Manual

LPWANminiUNI
SDI-12 and ModBus probes cloud logger ADVANCED

18.9.2024

Author: Ing. Petr Foltyn

Content

1. UPLINK

All characters within payload structure are HEX string unless otherwise stated. Whole string contains always pair of characters between 00 and FF if domain is not limited. Explanation is available in three numeric system:

1. HEX … characters are declared in following form: 0x00 to 0xFF
2. DEC … characters are described as usually 0 to 255
3. BIN … characters are described in following form: 0B00000000 to 0B11111111

1.1 NBIoT header

Payload within NBIoT infrastructure consists of 4 byte long unique ID (device ID) followed by information about battery voltage and signal strength. Then rest of the device specific data are transmitted. The device ID range is from 0x0000001 to 0xFFFFFFF which is defined by Solidus Tech s.r.o. in production process.

Byte	Meaning	Range	Note
1	MSB ID	0x00 to 0xFF	Unique ID of the device
2	2nd byte of ID	0x00 to 0xFF	
3	3rd byte of ID	0x00 to 0xFF	
4	LSB ID	0x00 to 0xFF	
5	Battery voltage	0x00 to 0xFF	Wagobox FF: Byte x 30 = voltage [mV] CamdenBoss FF: Byte/10 = voltage [V]
6	Signal quality	0x00 to 0xFF	0..63 – signal quality , 99 – signal quality not retrieved, RSSI = value -110 [dB]
7	Infobyte	0x00 to 0xFF	See below
8	Frame counter	0x00 to 0xFF	Payload counter

1.2 LoRaWAN header

There is only one byte in header available – battery voltage

Byte	Meaning	Range	Note
1	Battery voltage	0x00 to 0xFF	Wagobox FF: Byte x 30 = voltage [mV] CamdenBoss FF: Byte/10 = voltage [V]
2	Infobyte	0x00 to 0xFF	See below
3	Frame counter	0x00 to 0xFF	Payload counter

Info byte – 7th byte (NBIoT)/2nd byte(LoRaWAN)

MSB	6	5	4	3	2	1	LSB
HW revision	HW revision	HW revision	FW revision	FW revision	FW revision	Reserve	Downlink
0,1	0,1	0,1	0,1	0,1	0,1	X	0,1

2 SDI-12/ModBus descriptors

Descriptors define payload content based on connected probes with SDI-12 and ModBus interface.

2.1 SDI-12 descriptor

User can define particular probe presence and select required registers/data which will appear within the payload. Command probeX: should be followed with the information according the table below, separated by "," character and each probe descriptor should be terminated by "/" character:

Byte	Meaning	Range	Note
1	Device address	0x00 to 0xFF	0 to 9, a to z
2	Device type	0x01	SDI-12 interface
3	!aMx command register	0x00 to 0xFF	See below, if value = 0, then command aM! is used
4	Dx register requested	0x00 to 0x09	D0 to D9
5	Range of Dx register	0x00 to 0xFF	Mask of data nodes for given register

Meaning of 5th byte – Requested data from register Dx

MSB	6	5	4	3	2	1	LSB
Node 8	Node 7	Node 6	Node 5	Node 4	Node 3	Node 2	Node 1
0 = not req. 1 = required	0 = not req. 1 = required	0 = not req. 1 = required	0 = not req. 1 = required	0 = not req. 1 = required	0 = not req. 1 = required	0 = not req. 1 = required	0 = not req. 1 = required

Example: Trübner SMT100 SDI-12 probe should be in position 1 within the payload, SDI-12 address should be set to 1 and. Typical response of SMT100 probe is 1+21137+1.00+0.00+24.62+4.75 where node 1 is address, node 2 is uncalibrated raw data, node 3 is calibrated permittivity, node 4 is calibrated VWC, node 5 is temperature and node 6 is supply voltage. To get the data from the probe, 1M! command must be initiated and data then can be retrieved from register D0. Let's assume that we care only about VWC, temperature and permittivity data (node 3,4 and 5). Then format of the command will be:

```
probe0:1,1,0,0,1C/
```

explanations to the command structure:
probe0: - probe position in payload – first position is marked 0
1 – address 1,
1 – SDI-12 device type,
1 - 1M! command required to get the data,
0 – D0 register contains the data,
1C – here we convert the value to binary: 00011100 – node 5,4 and 3 are selected
/ - terminal character

2.1.1 Example with combination of SMT100 and TEROS 32

Let's assume that combination of SMT100 SDI-12 and TEROS 32 SDI-12 probes are required. SMT100 comes with address 1, TERROS 32 was assigned with address 2. SMT100 probe was explained in example above, so let's take the same setup. TERROS 32 data is described as follows:

Command	Response
This command reports instantaneous values.	
aM!	atttn
aD0!	a+<matricPotential>±<temperature>+<meta>
aD1!	a±<pitch>±<roll>
aD2!	a+<uPressure>±<temperature>

So the command aM! Will be used to retrieve the data and then data is available in registry D0 to D2. We will try to get all values from registry D0 and D2 except of address, we will skip register D1.
The desriptor commands will be as follows:

```
probe0:1,1,0,0,1C/…. descriptor for SMT100
probe1:2,1,0,0,6,2,6/ … descriptor for TERROS32
```

Let's break down the structure for TERROS 32:
probe 0: - probe position in payload – first position is marked 0
2 - address,
1 -SDI-12 device,
0 – 2M! command required to get the data,
0 – register D0 contains the data,
6 -> 0000 0110 – mask of node 2 and 3,
2 - register D2 contains the data,
6 -> 0000 0110 – mask of node 2 and 3,
/ - termination character

2.1.2 Example with multiple aMx! commands

Let's assume that user wants to connect the SDI-12 device where data is available under more than common aM! command. For example we will try to define following structure: aM! D0 register, nodes 3,4,5, then aM1!, D1 and D2 register, both nodes 2,3 and aM2! with D4 register and node 2 to 5. Address of the SDI-12 device is 3.

```
probe1:3,1,0,0,1C/2,1,6,2,6/4,4,3E/
```

Let's break the structure down:
probe1: - probe position in payload – first position is marked 0
3 - address,
1 – SDI-12 device,
0 – !3M command masked,
0 – D0 register contains tha data,

1C – node 3,4,5 masked
/ - description for aM! Terminated
2 – !3M1 commnad masked
1 – D1 register contains the data,
6 – Node 2 and 3 masked,
2 – D2 register contains data,
6 – Node 2 and 3 masked
/ - description for !3M1 terminated
4 – !3M2 command masked
4 – D4 register contains the data,
3E – mask for node 2 to 5
/ - description for !3M2 terminated

2.2 Data content encoding/decoding for SDI-12 interface

In most cases the response from different probes contains float data. To optimize payload length, the data will be always encoded to 3 bytes, where:

1st byte contains info about sign, exponent and exponent sign, see explanation below.
2nd and 3rd bytes are mantissa of the value rounded to fit 0 to 65535 range.

Meaning of 1st byte

MSB	6	5	4	3	2	1	LSB
Sign	Exponent sign	Reserve	Reserve	Reserve	Exponent	Exponent	Exponent
0 = positive 1 = negative	0 = positive 1 = negative	0,1	0,1	0,1	0,1	0,1	0,1

Examples:

0.02568 = 2568 e^{-2} => 2568 = 0x0A08, 1st byte = 0b01000010 = 0x42 => **0x420A08**
-688.28 = -6883 e^{-1} => 6883 = 0x1AE3, 1st byte 0b11000001 = 0xC1 => **0xC11AE3**

2.2.1 Example with real device response:

Let's assume that we have request from customer who requires data from following devices:
1.Tensiomark by Eco-Tech
2.SMT100 by TRUEBNER

Then let's create simple descriptors (see documentation for details):
1.probe0:0,1,0,0,6/
2.probe1:1,1,0,0,1C/

And here is commented result:
[2023-02-02_08:42:04:413]****NBIoT BC66 xmegaA4U DEMO***

[2023-02-02_08:42:04:413]First setting:5
[2023-02-02_08:42:04:668]*****************************
[2023-02-02_08:42:04:668]*****NBIoT SDI-12 logger*****
[2023-02-02_08:42:04:668]*****************************
[2023-02-02_08:42:04:668]FW version 3.3a
[2023-02-02_08:42:04:668]NBIoT xmegaXXA4U Kernel 2.1b
[2023-02-02_08:42:04:668]Built: Jan 31 2023,13:25:10
[2023-02-02_08:42:04:700]*****************************
[2023-02-02_08:42:04:747]Number of probes:1
[2023-02-02_08:42:05:064]*********Periodic mode********
[2023-02-02_08:42:05:639]PDP session renew:1/255
[2023-02-02_08:42:05:639]Probe descriptor:0
[2023-02-02_08:42:07:756]Probe0:
[2023-02-02_08:42:07:756]**0,1,0,0,6/ //Take first descriptor**
[2023-02-02_08:42:07:756]*****************************
[2023-02-02_08:42:07:756]Deparsing node 1
[2023-02-02_08:42:07:756]Probe address 0:
[2023-02-02_08:42:07:756]SDI command:0
[2023-02-02_08:42:08:120]ID: 011TENSIOMARK_2V0_10A28901 **//Probe identified**
[2023-02-02_08:42:08:347]Waiting time:8s **//Waiting time for measurement retrieved from the response of !0M measurement command**
[2023-02-02_08:42:17:370]Register D0 content:
[2023-02-02_08:42:17:370]0+23.10+6.576 **//D0 register content retrieved from SDI-12 bus**
[2023-02-02_08:42:17:370]SDI address:0
[2023-02-02_08:42:17:370]Measurement command: !0M
[2023-02-02_08:42:17:370]SDI register:D0
[2023-02-02_08:42:17:370]Mask:0x06 **//Value 2 and 3 should be involved to the payload**
[2023-02-02_08:42:17:370]*****************************
[2023-02-02_08:42:17:401]Node content:23.100
[2023-02-02_08:42:17:417]Info register0x:43
[2023-02-02_08:42:17:448]Exponent:-3
[2023-02-02_08:42:17:464]Mantisa:23099
[2023-02-02_08:42:17:479]Node content:6.576
[2023-02-02_08:42:17:646]Info register0x:43
[2023-02-02_08:42:17:646]Exponent:-3
[2023-02-02_08:42:17:646]Mantisa:6576
[2023-02-02_08:42:17:646]End of parser... **//Values deparsed and wrapped to the payload**
[2023-02-02_08:42:17:646]*****************************
[2023-02-02_08:42:17:646]Probe descriptor:1
[2023-02-02_08:42:19:706]Probe1:
[2023-02-02_08:42:19:706]**1,1,0,0,1C/ //next probe**
[2023-02-02_08:42:19:706]*****************************
[2023-02-02_08:42:19:722]Deparsing node 1
[2023-02-02_08:42:19:737]Probe address 1:
[2023-02-02_08:42:19:768]SDI command:0
[2023-02-02_08:42:20:192]ID: 113TRUEBNERSMT100038221217140034

[2023-02-02_08:42:20:450]Waiting time:2s
[2023-02-02_08:42:22:041]Register D0 content:
[2023-02-02_08:42:22:041]1+19148+1.03+0.00+22.16+7.93
[2023-02-02_08:42:22:057]SDI address:1
[2023-02-02_08:42:22:087]Measurement command: !1M
[2023-02-02_08:42:22:344]SDI register:D0
[2023-02-02_08:42:22:344]Mask:0x1C **//I care about node 3,4 and 5**
[2023-02-02_08:42:22:344]**********
[2023-02-02_08:42:22:344]Node content:1.030
[2023-02-02_08:42:22:344]Info register0x:44
[2023-02-02_08:42:22:344]Exponent:-4
[2023-02-02_08:42:22:344]Mantisa:10299
[2023-02-02_08:42:22:344]Node content:0.000
[2023-02-02_08:42:22:344]Info register0x:44
[2023-02-02_08:42:22:344]Exponent:-4
[2023-02-02_08:42:22:344]Mantisa:0
[2023-02-02_08:42:22:344]Node content:22.160
[2023-02-02_08:42:22:633]Info register0x:43
[2023-02-02_08:42:22:633]Exponent:-3
[2023-02-02_08:42:22:633]Mantisa:22160
[2023-02-02_08:42:22:633]End of parser...
[2023-02-02_08:42:22:633]**********
[2023-02-02_08:42:22:633]Payload:
[2023-02-02_08:42:22:633]FFFFFFFF76006400435A3B4319B044283B440000435690 **//we have total 5 values, each value is 3 bytes, see documentation how it's encryptet.**
[2023-02-02_08:42:22:633]SLEEPING...

And let's make reverse operation – check payload values and compare with real responses from the SDI-12 probes:

Probe Tensiomark response: 0+23.10+6.576
Probe SMT100 response: 1+19148+1.03+0.00+22.16+7.93
Payload: FFFFFFFF76006400435A3B4319B044283B440000435690
Let's break it down:
FFFFFFFF – Device NBIoT ID
76 – battery, 3540mV
00 – signal (demo, no signal)
64 – info byte
00 – frame counter
435A3B = 23.099
4319B0 = 6.576
44283B = 10.299
440000 = 0.0
435690 = 22.160

2.3 ModBus descriptor

ModBus probes provides different data structure in comparison with SDI-12. Typically we need register map, length of register and formula for calculating real values from register content. There are following probe types defined for ModBus interface:
3 – register length 2B
4- register length 4B (float registers)

Descriptor for ModBus probe has following structure:
probeX:A,PT,REG1,REG2,…,REGN/ where:
X – order within the payload
A – ModBus address
PT – probe type, see above
REG – register address
/ - termination character

Example: ModBus probe with register length 2B and ModBus address 1, registers 0x0001, 0x0002 and 0x0003 should be involved into payload on probe should occupy first position.

probe0:1,3,1,2,3/

Once the device starts contacting ModBus device there will be total 6 bytes (3 registers, 2B each) inserted into the payload. If ModBus probe is not reachable for some reason, the response will be 0xFFFF for each register.

Here is example of the terminal response for probe0:1,3,1,2,3,5,6/

```
Probe descriptor:0
Probe0:
1,3,1,2,3,5,6/
ModBus parser...
Reading ModBus device 1, register:0001
Error code:0
Register content:40C7
Reading ModBus device 1, register:0002
Error code:0
Register content:5A0F
Reading ModBus device 1, register:0003
Error code:0
Register content:AA84
Reading ModBus device 1, register:0005
Error code:0
Register content:0010
Reading ModBus device 1, register:0006
Error code:0
Register content:F01E
```

Probe descriptor:1
Probe1:
End of parser...
****
Payload length:27
Payload:FFFFFFFF6D00640040C75A0FAA840010F01E
SLEEPING in scan 60 min.

2.4 SDI-12 and ModBus probes combination

Combination of SDI-12 and ModBus probes are possible, please note that both probe types are powered from the same power unit, so it's necessary to select appropriate voltage which fits to both probe types. Following example combines 2 SDI-12 probes with one ModBus probe:

probe0:1,1,0,0,1C/
probe1:1,1,0,0,6/
probe2:1,3,1,2,3/

If for some reason the ModBus probe contain 2 or more different register length , for example 2B and 4 bytes, then use following definition:

probe0:1,3,1,2,3/
probe1:1,4,9,11/

2 different probe types were defined with the same ModBus address. The 2 bytes long registers will be inserted into the payload and then rest of the 4 bytes long registers.

2.5 Weather station sensors descriptors

Following table contains descriptors for non SDI-12 and non ModBus sensors which can be connected to the advanced logger including response involved within the payload.

Sensor	Interface	Descriptor	Payload contribution
T/RH/P in radiation shield, Connected to VCC,GND,SCL and SDA terminals	I2C	probex:1,7/	6 bytes total: 1st dual bytes = T/10 [°C] 2nd dual bytes = RH/10[%] 3rd dual bytes = Pa/100[hPa]
Rain tipp counter, connected to RAIN terminal	Counter	probex:1,8/	2 bytes total – incremental counter content
Leaf moisture, connected to LEAF terminal	Analog	probex:1,9/	2 bytes total – resistivity [ohms]
Wind*, Connected to WD – wind direction and WS – wind speed terminals	Analog Counter	probex:1,A/	3 bytes total: 1st byte = gust of wind 2nd byte = wind direction 3rd byte = average wind speed
Analog/Digital converter 4 channels, 0-3V**, connected to ADC1 to ADC4 terminals, see wiring	ADC	probex:1,B/	8 bytes total: 1st byte = MSB voltage CH1 2nd byte = LSB voltage CH1 3rd byte = MSB voltage CH2 4th byte = LSB voltage CH2 5th byte = MSB voltage CH3 6th byte = LSB voltage CH3 7th byte = MSB voltage CH4 8th byte = LSB voltage CH4

* please note that for gust of wind scaning purpose it's necessary to define sufficient scan time
** for CamdenBoss only

2.5.1 Analog/Digital converter (ADC) data interpretation

Each ADC channel contributes with 2B to the payload. To get the value in mV, multiply the value by 0.125. Example, CH1 contribution is 0x125C = 4700 => 4700*0.125 = 587.5mV

3 Command set

3.1 Advanced logger specific commands

Command	Decription	Default
probex:y	See detailed description above	-
initprobe	Clears all probe records in memory	-
probe?	Brings all defined probes descriptors to the terminal	
sensor	Brings measurement results to the terminal from all defined probes	-
newaddr:x	Changes address of connected SDI-12 device to a new value x. Please keep in mind that only one SDI-12 device can be connected in the same time when address request is sent	
ver	Brings version information to the terminal	-
sync:xx:yy	Time synchronization request, see further	Off
paralel:on	Turns on parallel mode, see further	
paralel:off	Turns off parallel mode, see further	Off
sdipwr:X	Defines output voltage for powering the probes, where X can be 5, 9 or 12 for 5V, 9V or 12V power output	5V
meastype:X	Defines which SDI-12 will be used for starting measurement, X is either M or C. If command C is used then only C! command will be initiated, rest of the measurement commands within the descriptor will be ignored	M
pwrmode:X	Defines if switched or continuous mode of powering should be used, X is either switch or cont. This selection is important for SDI-12 devices which require continuous power, then switch circuit is never turned on due to energy saving reasons. Continuous power can be taken either from the battery (Vcc) – nominal 3.6V or externaly while GND of the device and external power unit must be interconnected.	switch
mwt:x	Defines max waiting time to retrieve the results from the registry after measurement commands for SDI-12 bus. This is protection against eventual wrong waiting time retrieving from a probe response. The x parameter is in seconds.	99
wto:x	Defines waiting time offset where x is parameter is seconds. This parameters resolves eventual timing problem between measurement command send, waiting time obtained from a probe and measurement results availability within given D register.	0
tod:	Turn on delay where x is initial delay in s	0
wind	Brings values from the wind sensor to the terminal	-
rain	Brings rain counter to the terminal	-
adc	Brings values of voltage for CH1 to CH4 to the terminal	-
form:x	Change current form factor, x=WAGO for Wagobox, x=CB for CamdenBoss housing. This setting is important for battery content within the payload. For WAGO there is standard battery measurement, for CB there is 12V accu value of the voltage.	

parity:x	Defines parity for ModBus interface, valid values of x are no, even, odd.	even
baud:x	Defines baud rate for ModBus interface, where: x = 1 for 9600 x = 2 for 19200 x = 3 for 38400 x = 4 for 57600 x = 5 for 115200	

3.2 Kernel commands

Command	Decription	Default
reset	Restarts the device	-
sleep:x	Sets the interval between periodic transmissions, where x is time in minutes	60
scan:x	Sets the scan interval for checking alarm triggers values or collecting samples. If wind sensor is equipped, then due to the gust of wind detection and precise average wind speed calculation we recommend to keep scan parameter in 1 min.	1
show	Brings basic settings of the device to the terminal	-
ver	Brings version information and built date&time to the terminal	-

For details, see QuickStartNBIoT_BC66.docx documentation.

3.3 Network parameters change commands

Command	Decription	Default
netid:x	Sets a PLMN of the network which should be obtained from your provider and it's always country specific. If x is set to 0, then automatic network selection should work in most of the countries.	0
ipaddr:x	Defines the IP address of the server for receiving UDP uplinks from the device, where x is IP address in usual form e.g. 192.168.1.2	192.168.0.20
port:x	Defines the port of the server, where x is port number	4242
trxon	Forces the transceiver to wake up from the deep sleep in order to insert AT commands	-
AT	Ping to the transceiver	-
AT+QCGDEFCONT="IP","APN"	Sets the APN. If APN is not defined or definition is wrong, then invalid or none IP address is assigned and communication over NBIoT is disabled	m2mc.com
AT+QCGDEFCONT?	Checks the APN setting	-

AT+QBAND=x,y	Defines available bands for given provider, where x is number of bands and y is given band, for example AT+QBAND=3,3,8,20 means that 3 bands will be defined, B3, B8 and B20	2,8,20
AT+QBAND?	Checks the band setting	-
AT+QRST=1	SW restart of the transceiver. It's necessart to restart the transceiver each time when APN or bands are changed	-

Please note that the transceiver is forced to the deep sleep mode in 5s of inactivity, so the AT commands must be inserted quickly. Solidus Tech recommends using QCOM terminal for handling with NBIoT devices. This is Quectel product, however it can be downloaded from Solidus Tech repository here: https://iot.solidustech.cz/downloads/QCOM.zip

4 Time synchronization

If required, time synchronization feature can be activated. The device acquires the time stamp from the network and based on command sync:xx:yy parameter will synchronize next transmission to required time where xx are hours, yy are minutes. For example, if sync:14:00 is inserted, then next and further transmissions will be synchronized to 14:00 time. Let's assume that we have sleep time = 120 min. and scan time = 120 and the battery was inserted 12:37 to the device. The device will start, once it's connected it obtains the timestamp, set the synchronization time, makes measurement (depends on connected probes, it can take from few seconds up to few minutes) and makes first transmission let's assume 12:38. Next transmission is then at 14:00, another one at 16:00 etc. Every 24 hours the time will be checked and eventually synchronized.
Parameter x can be set in range 0 to 23, y can be set in range 0 to 59, if y = 255, then synchronization feature is turned off and the device sends payloads within the time frame corresponding to inserted battery, sleep and scan time parameters.

5 Parallel measurement

User can decide to enable or disable parallel mode of sending initial SDI-12 measurement commands which are contains in first node of the probe descriptor. Parallel mode can save significant amount of energy especially when the device should work with more probes where waiting time for result is relatively long (e.g. sap flow sensor – 92s). Parallel mode sends all aMx! commands for requested probes and gets maximum waiting time, then it waits for results and then retrieves all requested register content from all associated probes. This approach can significantly reduce power consumption, because waiting time for result is not cumulated, but it's implemented based on maximum waiting time only once.

6 Downlink

6.1 NBIoT

All serial UART commands are available over downlink channel, see NBIoT downlink documentation

6.2 LoRaWAN

A0XXXXYYYY - Timing change command, where XXXX is sleep time in minutes – periodic message sending and YYYY is scan time in minutes for alarm levels scaning. Because this kind of device contains no alarm monitoring, make sure that XXXX=YYYY. For example, if you want to change periodic time to 60min. use following command structure: A0003C003C.

7 SDI-12 address change

Please note that the probes must be addressed properly and it requires particular setting if the probes are not delivered by SolidusTech and labeled with right address. The device supports the probes with address 0 to 9 and a to z. You can either use external tool for address setting or you can use delivered device to do it. For this purpose, the device must be connected to the PC over USB/UART and proper terminal program must be installed, see Quick start guide for given device. Once the prerequisites are met, follow this instruction:

1. Connect the probe in which you need to change the address
2. Check that the probe is visible by typing **sensor** command, then you should see terminal response

```
Probe address 1:
ID: 1SMT100 SDI-12
register content:1+11496+15.41+28.21+28.29+5.02
address:1
Soil Moisture:28.21

Temperature:28.29
Permeability:15.41
Probe voltage:5.02
```

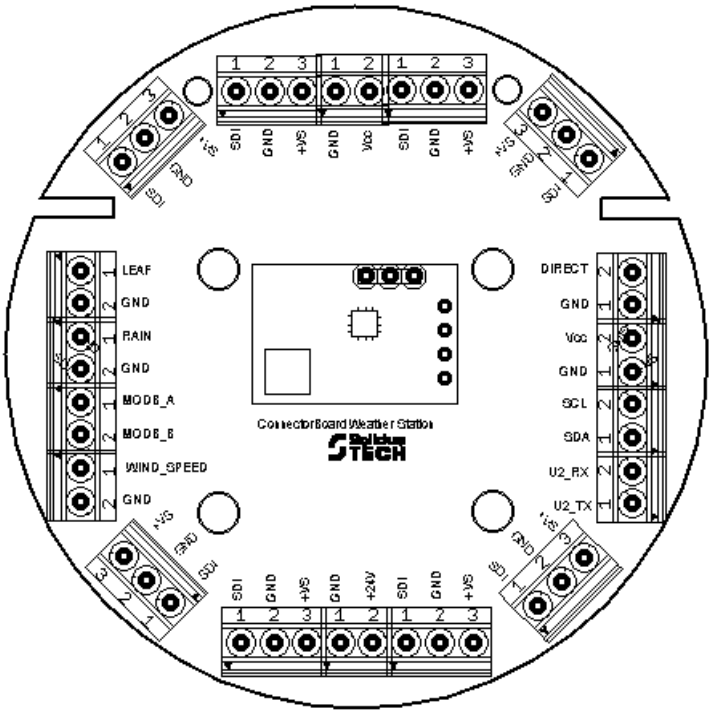
3. with probe identification and measurement results
4. Use **newaddr:x** command where x is the desired address
5. Check again over sensor command to make yourself sure that the address was changed.

8 Wiring

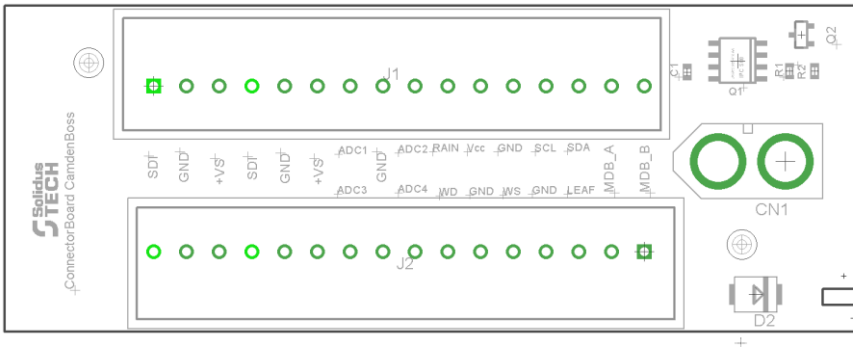
Up to 10 SDI-12 probes can be connected to the device – depends on specification upon order. The device supports bus configuration so all probes are connected in parallel order. On each device you will find a single screw connector (green colour) – typically for 1 and 2 probes, or nest of blue screw connectors for connecting 4 probes. Each connector is marked with letters BWG (or S+I) which stands for wire colors leading from SMT-100 probes.

B – Brown (+) - +5V
W – White (I) - GROUND
G – Green (S) – SDI-12

8.1 Connector ring board – TOP VIEW:



8.2 CamdenBoss board – TOP VIEW:



9 Revision

1.0	Initial FW
1.1	• Parallel measurement implemented • Scan time bigger than 256min parameter corrected
1.2	• Time synchronization added • Parallel measurement added
1.3	• ModBus support added • Selectable output power added, sdipwr:X command
1.4	• mwt and wto commands added
1.5	• tod parameter added
1.6	• CamdenBoss form factor from single 12V accu powered • 12V accu measurement instead of 3.3V nominal voltage for CamdenBoss • New device descriptor for external ADC added
1.7	• Parity and baud rate is configurable for ModBus • Battery interpretation for Wagobox form factor corrected to original interpretation • Float registers support added for ModBus probes