PN: SS-HDC2010+CCS811#I2C   SKU: ITBP-6006

s-Sense by itbrainpower.net

CC811 + HDC2010 I2C sensor breakout

Equipped with **CCS811** (CO2 and tVOC) and **HDC2010** ((temperature and humidity) sensors. This sensor bundle it is probably the best **air quality** sensor bundle available worldwide.

CCS811 + HDC2010 breakout features:
- *I2C interface**, 3.3-5V auto* [default, the I2C reference voltage follows the power in voltage],
- *I2C reference voltage* option via solder jumper [Eg.: power at *5V and I2C at 3.3V*; *power and I2C at 2.2V*],
- *alternative sensor I2C addresses* via solder jumpers,
- power in with voltages between 3.3-5V [default],
- big soldering pads on 2.54 mm [0.1"] pitch for standard row header / connector,
- very light and slim - 13x27mm, around 2grams,
- compatible with Arduino [all 3.3V and 5V], xyz-mIoT shields, Raspberry PI [all versions] and other SBC / MCU Arduino [Beagle Bone, Teensy, ...]
- Arduino C, Raspberry PI [Python] software support.

CCS811 sensor features:
- equivalent CO2 range is 400ppm up to 29206ppm
- equivalent tVOC (Total Volatile Organic Compound) output range is from 0ppb up to 32768ppb
- ambient temperature range for operation -40 ~ +80C

CCS811 datasheet: itbrainpower.net/downloadables/CCS811_DS000459_5-00.pdf

HDC2010 sensor features:
- relative humidity range 0% - 100%
- humidity accuracy +-2% (factory calibrated)
- temperature from -40C to +125C (functional) and -40C to +80C (operational)
- typical temperature accuracy +-0.2C (factory calibrated)
- embedded heating element to burn away condensation

HDC2010 datasheet: https://itbrainpower.net/downloadables/hdc2010.pdf

Designed and manufactured in EU by R&D Software Solutions.

CCS811** + HDC2010** breakout

carbon dioxide, total Volatile Organic Compounds, temperature and humidity - air quality sensor

PN: SS-HDC2010+CCS811#I2C   SKU: ITBP-6006

* Additional signals are available on PCB bottom:
- CCS811 WAKE on PAD6, CCS811 RESET on PAD7 and CCS811 INTERRUPT on PAD8.
- HDC2010 INTERRUPT signal it is available on PAD9.

** digital sensors provided by AMS (CCS811) and Texas Instruments (HDC2010)

CCS811 + HDC2010 breakout :: INTERFACES, PADS and SOLDER JUMPERS

I2C and POWERING INTERFACE

In the left edge of the top PCB side, top to bottom:
1. Vin - POWER IN 3.3V / 5V (default) *
2. 3V3 - Vdd - LDO output (default) / LDO bypass IN **
3. SDA - I2C SDA [default, pulled up at Vin]***
4. SCL - I2C SCL [default, pullup up at Vin]***
5. GND - POWER and I2C GROUND

* Default, the I2C reference voltage follow the Vin voltage. If your board (Arduino/other) it's 5V compliant, connect +5V to Vin. Similar, for 3.3V compliant board (RPI/Arduino) connect 3.3V source to Vin.
** 3.3V voltage it's available on Vdd PAD, if Vin it's connected to 5V. Bypass LDO - see schema and read docs.
*** Default, the I2C reference voltage follows the Vin voltage! Read bellow about I2C reference voltage and *s1* solder jumper.
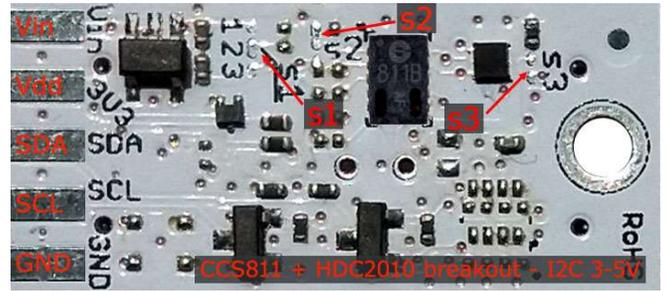
Image 1 – sensor top side

SOLDER JUMPERS

s1 : I2C reference [Vin or 3V3-bypass], default Vin *
s2 : CCS811 I2C address [0x5A/0x5B], default 0x5A *
s3 : HDC2010 I2C address [0x40/0x41], default 0x40 *
* Read about s1, s2 and s3 functionality bellow.

SIGNALS INTERFACE and MARKING

In the left edge of the bottom PCB side, top to bottom:
6. CCS811 WAKE* input (active LOW)
7. CCS811 RESET* input (active LOW)
8. CCS811 INTERRUPT* output (w. pullup***)
9. HDC2010 INTERRUPT** output (w. pullup***)
nc. NOT CONNECTED

* In order to wake the CCS811 sensor, the WAKE signal must be shifted and kept at LOW level (connect directly to GND or controlled via MCU digital port).
* Read about CCS811 WAKE, RESET and INTERRUPTS in CCS811 documentation!
** Read about HDC2010 INTERRUPTS in HDC2010 documentation!
* & ** Taking a look at how-to posts and inside software examples may be quite helpful.
*** Default HIGH level.
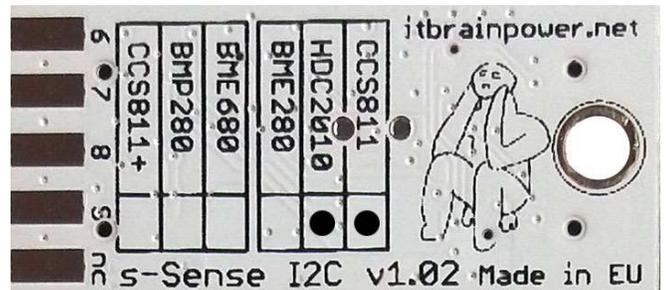
Image 2 – sensor bottom side

Observe the sensor breakout marking on PCB board (here, CCS811 and HDC2010 sensors are marked as populated).

CCS811 + HDC2010 breakout :: Arduino / Raspberry PI basic interfacing

CCS811+HDC2010 breakout	Arduino 5V [Eg.: UNO]	Arduino 3.3V [Eg.: Zero]	RPI [B/B+/II/3/3+/Zero/Zero W]
Vin PAD*	5V	3.3V	PIN1 (3.3V)
Vdd PAD (3V3)*	-	-	-
SDA PAD	SDA	SDA	PIN3 (SDA)
SCL PAD	SCL	SCL	PIN5 (SCL)
GND PAD	GND	GND	PIN9 (GND)
6 - CCS WAKE **	D5 / GND	D5 / GND	PIN9 (GND)
7 - CCS RESET ***	-	-	-
8 - CCS INT ***	-	-	-
9 - HDC2010 INT ****	-	-	-

* here, power in default mode (sensor & MCU power and I2C at same voltage!!). For alternate mode see solder jumper description bellow.

** PAD6 - CCS811 WAKE, when connected to GND, the CCS811 will avoid enter into SLEEP mode [sensor it's ACTIVE].

*** CCS811 RESET (input) and INT (output) signals are default HIGH level. Active state, for both signals, in LOW level. Read CCS811 docs.

**** HDC2010 INT (output) signal is default HIGH level and the behavior it's programable. If required, connect to one GPIO [set as INPUT], INTERRUPT capable. See examples and read HDC2010 docs.

CCS811 + HDC2010 breakout :: solder jumpers description; power scenarios

s1* - I2C reference voltage selector - shipped with circuit 1-2 closed and circuit 2-3 open (default mode).

In **default mode**, the I2C reference voltage follows the voltage applied to Vin! Default mode use cases:
    - shield/MCU it's 5V compatible ==> connect Vin at 5V; the I2C reference it's 5V!
    - SBC/shield/MCU it's 3.3V compatible ==>connect Vin at 3.3V; the I2C reference it's 3.3V!.

Alternate mode - I2C reference voltage follows the Vdd voltage! - modify *s1* as: 1-2 circuit open, 2-3 circuit closed [carefully interrupt/cut the circuit line between pads 1 and 2, and solder over pads 2 and 3].

Alternate mode scenarios:
    - apply 3.6-7V to Vin. The I2C reference voltage it's set at 3.3V! Also, at Vdd pad (LDO output) 3.3V (up to 100mA) it's available for third part powering (anyway, try to keep the sunk current as low as possible, in order to avoid thermal errors).
    - keep Vin not connected. Apply 2.8 or 3.3V to Vdd (the LDO it's bypassed). The I2C reference voltage it's set at voltage applied to Vdd pad!

s2 - **CCS811 I2C address** - default circuit open, CCS811 having 0x5A as I2C address. Solder over *s2* pads [circuit closed] in order to select the alternate I2C address [0x5B].

s3 - **HDC2010 I2C address** - default circuit open, HDC2010 having 0x40 as I2C address. Solder over *s3* pads [circuit closed] in order to select the alternate I2C address [0x41].

* Consult s-Sense block schema for more information. See projects/how to section.

CCS811 + HDC2010 breakout :: ARDUINO & RaspberryPI software

https://itbrainpower.net/downloads.php#s-Sense

CCS811 + HDC2010 breakout :: ARDUINO & RaspberryPI documentation

https://itbrainpower.net/downloads.php#s-Sense_documentation

CCS811 + HDC2010 breakout :: projects and how to

https://itbrainpower.net/projects

CCS811 + HDC2010 breakout :: part number / SKU

PN: SS-HDC2010+CCS811#I2C   SKU: ITBP-6006