Manual for EMP2PC

DEVICE : EMP2PC

FEATURES : Converter for EMP-sensor to serial communications over USB

APPLICATIONS : Detecting / monitoring lighting and other types of high energy discharges

NOTE : This device requires programming skills, for data-processing (plot diagrams, store data and / or data-transmission over internet)
SPECIFICATIONS

PC connection: USB
Current: 100mA @ 5V DC
Communication: Serial (by SiLabs CP2102)
Transfer-speed: 115.200 Kbps
Data-parameters: 8-N-1
Dimensions (module A): 1.98 x 0.76 x 0.43 inch / 50.2 x 19.3 x 11.0 mm (L x W x H)
Weight (module A): 0.17 Oz / 4.7 gram
Dimensions (module B): 1.98 x 0.76 x 0.67 inch / 50.2 x 19.3 x 17.0 mm (L x W x H)
Weight (module B): 0.31 Oz / 8.9 gram

Length of data-string: 20 measurements per transmission (Sample-speed = Low)
100 measurements per transmission (Sample-speed: Medium, High and Highest)

Type of values: Present measurements in HEX or Decimal-values in string of 20 or 100
values per transmission or so called "burst".

Good to know: Decimal values requires an internal conversion and results into a slower sampling-rate. The speed can be tested by a command.
INFO

Why

An EMP-sensor is a device for detecting events of high energetic magnetic pulses. Our original EMP-sensor (V2) got a LED and audible alarm onboard. By these, a user can be warned quickly by sound and light-signals when an EMP-events occur. These sensors can also be used for a variety of electronic-projects.

However, under certain circumstances, warning-sounds and -blinks are not enough. Especially if analysis of EMP-events are required at computer-level, like:

- Plot / analyze data, nearly realtime at connected or remote computers
- Study (more) detailed characteristics about EMPs

A computer can only operate by digital data. This means a conversion between sensor and PC is needed.
If you’re not familiar with electronics, or prefer to go for a quick solution... the EMP2PC is the best option to get this job done. This module will convert EMP-sensor-data straight into digital chunks of data and transmit it to a computer over a USB-port. Besides, this module offers several flexible settings and can be controlled by computer, like:

- Samples per second (from 100 upto 1500 / sec)
- Detection-level
- Digital output (HEX of decimal)
- Detecting sensor

Available versions

Choice

If you prefer to solely experiment with an EMP-sensor, get it. Play with it, experiment with it or use it for a project. If at a later moment you like to go analyzing data: go for Module A. Insert the previously bought EMP-sensor and you’re ready to go.

For those who are not interested in playing around or experimenting with electronics, but like to get knee-deep into “data-mining” right away: go for Module B.
THE SHORT ROUTE
OK., we get it. You're the smart guy / girl, who don't like spent (too) much time reading.

Fine, here we go :

INSTALL
Step 1 : Download driver for SiLabs CP2102
Step 2 : Insert EMP2PC in available USB-port of PC
Step 3 : Computer complains about driver ?
    Yes => Install downloaded driver
    No => Go next step
Step 4 : Check which COM-port is set
Step 5 : Set communication-parameters of driver to 115.200 Baud, 8N1
Step 6 : Start up a program to communicate with EMP2PC (*)
Step 7a : Done ! Go to COMMANDS (see below)
Step 7b : Oops, something went wrong.
    Remove EMP2PC USB-stick, reboot PC and start over from step 2.
    If failure maintains, well, Google might be your friend,
    or..... read the following pages (more) carefully.

(*)A load of freeware, shareware and even scripts are available at internet. Even Python-scripts for plotting, straight from COM-Port data.
A general used program is Putty. We tried... but swiftly moved to CoolTerm. Great tool during development and testing ! It offers text INPUT and OUTPUT split apart. When receiving over 1.000 data-samples per second, you'll start to figure out why.

COMMANDS
No full explanation here. After all, you went for the short way...
At command-prompt, for help just enter H, h or ? and you get the full list. That should do it.

Good to know : measured values are between 0 to 1023. A low value (0) means no activity at all. And a value of "1023" refers to a serious EMP-event which took place (maybe even pretty close to you...).

Enjoy !
THE LONGER ROUTE

Software
Currently there's no software developed for the EMP2PC USB-stick. However, to get up and running at your PC you probably need 2 things:

1) Driver for your Operating System (Microsoft Windows, MacOS, Linux, whatever) for SiLabs CP2102
2) Software for data-communication over (Virtual) Serial Port.

This might look confusing when not familiar with it.
To make it clear(er) to understand : the driver creates the highway, between PC and EMP2PC. The Data-acquisition-software makes it possible to send commands to the EMP2USB and -likewise-retreiving data, like statuses and measurements of the EMP-sensor.

Step 1 (Driver)
The EMP2PC USB-stick contains a SiLabs CP2102 for serial communications. When inserting the USB-stick into your computer, it will search for a driver. Some PC's got this driver already installed. Some don't and a minor error-message may pop-up. Great!
In this case, it's time to go internet-surfing. Go for www.silabs.com, or just go for a search-engine like yahoo.com. Enter the following phrase "SiLabs CP210x VCP driver". For Yahoo, you can also cut and paste the following link: http://lmgtfy.com/?q=CP2102+drivers+download+SiLabs
Very likely you will be presented a link, where you can download several versions, for Windows, MacOS, Linux or....
Download the most recent version, install it, and you solved a minor problem.

NOTES:
- VCP stands for Virtual Com-Port
- Set the communication-parameters to : 115.200 Baud (Bps) and 8N1 (*)

(*) See Extra A, to see how to get this done in Windows 7

Step 2 (Data-communication)
Now it's time to "talk" to the module by PC, like sending commands and retreiving data.
At the internet there are tonnes of software available, in order to communicate over a serial (COM)-port.
Even free downloadable scripts in Python and other scripting languages can be found to even plotting graphs.
Besides, some users prefer to import data straight into a program they created themselves, other prefer plotting a graph, and.. The list keeps going on. For this reason we didn't make any software for the module. A user or community should be able to do it his/herself. In fact, we believe it will result in even more sophisticated solutions than we could think about.

If you're not able to write code, you can go for Putty or CoolTerm. We used CoolTerm during development and testing. It's a nice tool, which makes it possible to even store received data while keeping INPUT and OUTPUT separate. This feature is very handy, especially if you receive a lot of data per second and still want to be able to send a command to the module.

See Extra B about to get started under CoolTerm. We choose CoolTerm, because it's freeware and available for Mac, Windows and Linux.
<table>
<thead>
<tr>
<th>Command</th>
<th>Meaning</th>
<th>What it does</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Start</td>
<td>Start sampling data</td>
</tr>
<tr>
<td>-</td>
<td>Stop</td>
<td>Stop sampling data</td>
</tr>
<tr>
<td>O</td>
<td>Overview</td>
<td>Shows the actual status of the module (*)</td>
</tr>
<tr>
<td>D</td>
<td>Detect</td>
<td>Detecting if EMP-sensor is present or missing (**)</td>
</tr>
<tr>
<td>T</td>
<td>Test</td>
<td>A sampling-rate test. If no EMP-sensor present, test will be cancelled. A test will take between 1 and 2 seconds. Data will be transmitted to computer during test. If test took place while sampling (+) was active, further sampling will be stopped (-) after this test.</td>
</tr>
<tr>
<td>S</td>
<td>Speed</td>
<td>Shifting to the desired speed, from Low, Medium, High, Highest.</td>
</tr>
<tr>
<td>L</td>
<td>Level</td>
<td>Trigger-function to start sampling. When set to 0%: continuous sampling</td>
</tr>
<tr>
<td>V</td>
<td>Value</td>
<td>How data will be transmitted to computer. Data can be presented as decimal (DEC) or hexa-decimal (HEX) values. By default, the decimal values is selected. When using this command, you shift between HEX and DEC. NOTE: when HEX is selected, the sampling-rate will be higher.</td>
</tr>
<tr>
<td>R</td>
<td>Reset</td>
<td>Set the module back to default parameters.</td>
</tr>
<tr>
<td>?</td>
<td>Help</td>
<td>Shows available commands.</td>
</tr>
</tbody>
</table>

(*) Command "O" shows the status of the module, concerning the following parameters:

**EMP Sensor**
- Shows if a EMP-sensor is detected, like "Present" or "Missing".

**EMP-Logging**
- Shows the actual task of the module, concerning sampling data for the EMP-sensor.
  - **Idle**
    - The module is not sampling
  - **Active**
    - The module is sampling data and transmitting results to a computer
  - --
    - EMP-sensor or missing, which means it can't perform a task / sampling

**Sampling-rate**
- How fast the module is sampling data from an EMP-sensor.
  - **Slow**
    - Sampling-rate: 100 samples per second (approx.)
    - HEX / DEC: almost equal sampling-rate
    - Transmit: 20 samples sent per message (to computer)
    - Sampling period: about 200 milli-seconds.
  - **Medium**
    - Sampling-rate: 550 to 650 samples per second.
    - HEX / DEC: HEX is faster (650)
    - Transmit: 100 samples sent per message (to computer)
    - Sampling-period: Roughly 170 milliseconds

See next page for other speeds
- **High**
  - Sampling-rate: 900 to 1100 samples per second.
  - HEX / DEC: HEX is faster (1100)
  - Transmit: 100 samples sent per message (to computer)
  - Sampling-period: Roughly 100 milliseconds

- **Highest**
  - Sampling-rate: 1100 to 1500 samples per second.
  - HEX / DEC: HEX is faster (1500)
  - Transmit: 100 samples sent per message (to computer)
  - Sampling-period: Roughly 75 milliseconds

**EMP-Level**
The minimum level (in percentage) at which level sampling should be started.
When no activity is detected, it will result in a value zero (0). When activity is measured, the value will increase to a maximum value of 1023. If you prefer to select a trigger-like function, you can modify the detection-level. The net result is, the module will only start to measure at pre-set level (or value). This measuring will take place during a given (sampling) period. If the EMP-activity is still beyond this value, it'll continue for another (sampling) period.

- The lower the EMP-level, the faster the trigger will start measuring.
  By this, it becomes more clear why a value of "0%" will result in continuous sampling.
- The higher the sampling-rate is set, the more detailed info can be obtained about an EMP-event. However, it also means a more intense burst of data for your computer to execute.

(***) "D" will detect if an EMP-sensor is present.
This function is actually ment for module A, but will also work for Module B. The module can detect if an EMP-sensor got removed / disconnected. If this happens, a repetitive error-message will be sent to the computer. Also, this can be remotely verified by using command "D", when no sampling took place.
There’s an important situation to be aware of. This is in case the module is inserted in an USB-port, while an EMP-sensor is missing. If, in this situation, an EMP-sensor will be connected to the module, a reset of USB-port might be required. This actually means, to remove the module from USB-port, attach EMP-sensor, and re-insert module in USB-port. In other words, this function is actually ment to detect if something is going on / wrong, while the module was equiped with an EMP-sensor.

"Q"-command
This is a reserved feature. It is ment for computers, running software, sniffing each available COM-port on a regular basis, if a module is present. If detected, the module will return the text "ACK". In case a computer expects data, but there’s a failure, the computer can detect where the problem is located: missing or defective EMP-sensor, or missing module.
**Extra (A)**

Set parameters for CP2102 in Windows 7

We only got experiences about how to go for the proper settings under Windows 7. It will be described, including screenshots.

**Step 1**
Select "Start"-button.

![Start button](image)

**Step 2**
Select "Computer" with right mouse button. A pop-up screen with options will be shown.

**Step 3**
Select tab "Properties". Now you get the menu for "Computer Management"

![Computer Management](image)

**Step 4**
Select "Device Manager". At the right part of the menu, some hardware-devices are shown.

**Step 5**
Select "Ports (COM&LPT)". A sub-menu will be visible, presenting which hardware is connected.

**Step 6**
Select "Silicon Labs CP210x USB to UART Bridge (COMx)"
The "X" represents a number. "Silicon Labs CP210x...." refers to you EMP2PC module.

**Step 7**
Right click this item and another pop-up menu will be visible.

![Silicon Labs CP210x USB to UART Bridge](image)

**Step 8**
Select tab "Properties". The next menu we get, is the one what it's all about.

**Step 9**
Select "Port Settings"-tab.

**Step 10**
Select all parameters as shown in picture. *(115200, 8, None, 1 and None)*

**Step 11**
Finally select OK and you're ready.

![Port Settings](image)

If all went well, the hardware-part is done. The computer is now able to communicate with the module.
Extra (B)
CoolTerm, getting started

When CoolTerm is installed, a clickable icon may be available at the main screen.

Double click this icon and CoolTerm will load. When loaded, the main menu will be shown.

There's yet no communication available between your computer and the module. For this, select the "Options"-button. A new menu will pop-up, to select parameters. At the left side of the screen, there are 2 main important links: Serial Port and Terminal.

Default "Serial Port" is active. At the right, you see an example and parameters to be set.
Port : Select COM-port the module is located
Baudrate : Select 115200
Data Bits : 8
Parity : None
Stop Bits : 1
Besides, only options DTR On and RTS On, should be selected.

Now select "Terminal" and select "Line-Mode" and click "OK"-button.
Next, click "Connect"-button, to set-up a communications-link to the module.

When connection is ready, you probably will see 2 blank fields, while RTS and DTR will be active, as can be seen below. The fields are for:

- OUTPUT (upper field) will show data or results, sent from module to computer
- INPUT (lower field) is for entering commands from computer to module.

To start, enter character "O" and push ENTER-button. This will transmit command "O" to the module. As a result you should get the current module-status in the upper (OUTPUT) field.