

UPS_Plco_0x58_18_01_2016.hex recent changes

Added Features, Bug fixes

1. Added mirror variable to 0x69 -> UPS Plco Module Status Registers, located at (14 or 0x0E) with name **Pico_run**

Explanation:

It is a 16 bit unsigned variable that its value is changing every 10 ms within the main loop. Reading two times of this variable must return a different value (with interval longer than 10 ms), if not, means that system hangs-up, and need to be reset, if not restarted by other Plco protection internal mechanism (watch-dog, and supervising watch dog). As these Plco protection mechanisms are always restarting the system when something goes wrong. Reason of existence of this variable is just to confirm the remote user that everything is working well and give feedback to the remote user that system is running properly. As it is a mirror variable, writing to it nothing change, will be again re-written with the newer internal value.

Usage:

```
sudo i2cget -y 1 0x69 0x0e w
```

or

```
sudo i2cget -y 1 0x69 0x0e w && sudo i2cget -y 1 0x69 0x0e w
```

Must return every time when read different value every time. If read and compared, if equal system hang-up, if not equal, system is running properly. However during normal operation it should never happen. It is just for remote users to be sure that system is working properly

Example:

```
sudo i2cget -y 1 0x69 0x0e w && sudo i2cget -y 1 0x69 0x0e w
```

0x15b2

0x15c1

Simple python script can be done by user to do this testing automatically.

2. Reviewed the internal watch-dog, so now it is more sensitive and not excuse any delays in execution, as a result if any delay in main loop happen (a long list of not executed interrupts, will reset the Plco). Plco software recognize the reason of the reset, and it is doing different booting process, without setting RTC variables set to 2000, and 1, 1, 1 as it is default.

3. Second level of watch-dog has been implemented. The Plco exams now the system status every second and if conditions not met, restart the Plco.

4. Improved the factory setup. In some cases a hand initiated factory setup has been required. Now, after firmware update, system always has a proper factory setup.

5. Improved hardware reset. If UPSR button will be pressed (hardware reset of the UPS Plco) the RTC will continue running (interrupted for a very short time, which is not affecting the RTC), without losing or resetting the RTC time values.

6. Added information over RS232 that is printing on terminal (i.e. minicom) when charger is ON or OFF. Requested to have activated RS232 on the Plco if RS232 is used, default RS232 is OFF.

7. Added variable that can be read by user (remotely) to see if charger is activated or not

Usage:

```
sudo i2cget -y 1 0x69 0x10
```

0x01 - means Charger is ON

If the CHG LED is not lit, means that charger is ON, but battery is fully charged

0x00 - means Charger is OFF

8. Rewritten and significantly improved the battery protection system. Plco is now not allowing deep discharge of the battery on any case. The threshold of the cut-off battery is now 3.15V. It is activated always if battery level is lower than this one.

9. Improved the picofu.py to picofu3.py with better factory request, so always after system firmware update Plco has a default values for their internal variables

10. Improved the handling of the KEYA and KEYB, now when pressed, must be reset to be sensitive again. So, beep is audible only once.

Usage:

`sudo i2cset -y 1 0x69 0x09 0` reset the KEYA if preset, and make it again available for use.

This new implementation gives time to the software running on the Raspberry Pi to read the KEY status