

Пример использования SNMP на ПАК «МАЛИНА»

Поскольку в настоящее время не существует стандартных объектов MIB для работы с инверторами МикроАрт, необходимо воспользоваться текущими возможностями расширения NET-SNMP-EXTEND-MIB::nsExtendObjects

Необходимо войти в консоль Raspberry Pi. Локально или через SSH

Пользователь: pi

Пароль: raspberry

1. Устанавливаем клиент и сервер:

Код:

```
sudo apt-get update
sudo apt-get install snmp snmp-mibs-downloader
sudo apt-get install snmpd
```

Сервер нужно запустить: `service snmpd start`

Проверить syslog и статус `service snmpd status`

Код:

```
root@monitor1:/home/pi/malina/malina# service snmpd status
[ ok ] snmpd is running.
```

При необходимости добавить snmpd в автозапуск. На Малине работает chkconfig (кто привык)

2. Примеры конфигов /etc/snmp/* на гитхабе:

<https://github.com/Osolemio/malina/tree/master/other/etc/snmp>

В конце файла snmpd.conf добавляются расширения - запросы состояния МАП, МРРТ и батареи.

Код:

```
extend map /usr/bin/php /var/www/html/net_map.php
extend mppt /usr/bin/php /var/www/html/net_mppt.php
extend battery /usr/bin/php /var/www/html/net_battery.php
```

И обратите внимание на ваш IP адрес. Community name и прочие чувствительные для вас настройки.

Проверьте насколько скрипт net_battery.php соответствует последней версии на GitHub.

<https://github.com/Osolemio/malina/tree/master/html>

Только последняя версия файла будет работать корректно при вызове из командной строки

Затем, в командной строке вам необходимо посмотреть работу всех ваших переменных

Код:

```
snmpwalk -v1 -c public localhost NET-SNMP-EXTEND-MIB::nsExtendObjects
```

.....

```
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."map" = STRING:
{"time":"23:39:02","_MODE":"13","_Status_Char":"0","_Uacc":"52.1","_Iacc":"0",
"_Pload":"0","_F_Acc_Over":"0","_F_Net_Over":"0","_UNET":"232","_INET":"5",
"_PNET":"1000","_TFNET":"125","_ThFMAP":"125","_UOUTmed":"230","_TFNET_Limit":
"125","_UNET_Limit":"266","_RSErrSis":"0","_RSErrJobM":"0","_RSErrJob":"0",
"RSWarning":"0","_Temp_Grad0":"10","_Temp_Grad2":"18","_INET_16_4":"4.5","_Iac
c_med_A_u16":"0.0","_Temp_off":"10","_E_NET":"441925","_E_ACC":"106741","_E_AC
```

```
C_CHARGE":"6253","_Uacc_optim":"0.0","_I_acc_avg":"0.0","_I_mppt_avg":"0.0","
_I2C_Err":"0","_Temp_Grad1":"20","_Relay1":"1","_Relay2":"2","_Flag_ECO":"0",
"_RSErrDop":"0","_flagUnet2":"2"}
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."mppt" = STRING:
{"time":"23:39:02","Vc_PV":"51.7","Ic_PV":"0.0","V_Bat":"52.1","P_PV":"0","P_
Out":"0","P_Load":"0","P_curr":"0","I_Ch":"0.0","I_Out":"0.0","Temp_Int":"14"
,"Temp_Bat":"10","Pwr_kW":"0.000","Sign_C0":"0","Sign_C1":"0","I_EXTS0":"0","
I_EXTS1":"0","P_EXTS0":"0","P_EXTS1":"0","Relay_C":"0","RSErrSis":"0","Mode":
"B","Sign":"+","MPP":"L","windspeed":"65535"}
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."battery" = STRING:
{"number":"638192","date":"2016-12-
07","time":"23:38:17","integral_dCdt":"0.000","C_current_Ah":"0.000","C_curre
nt_percent":"100.000","I_avg":"0.00","user_counter":"192.432","estimated_SOC"
:"100.0","ah_accumulator":"0.000","timer":"23361"}
.....
```

Если вы это видите – все хорошо!

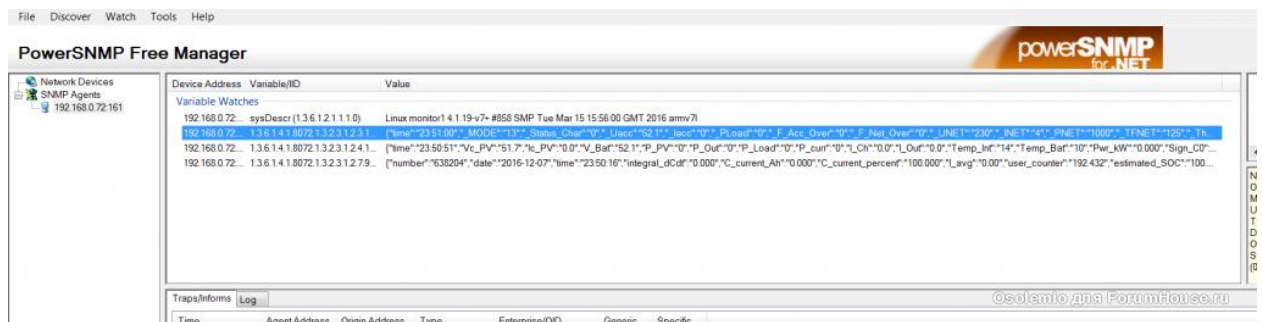
Теперь нужно найти номера расширений:

Код:

```
root@monitor1:~# snmptranslate -On NET-SNMP-EXTEND-
MIB::nsExtendOutputFull.\"map\"
.1.3.6.1.4.1.8072.1.3.2.3.1.2.3.109.97.112
root@monitor1:~# snmptranslate -On NET-SNMP-EXTEND-
MIB::nsExtendOutputFull.\"mppt\"
.1.3.6.1.4.1.8072.1.3.2.3.1.2.4.109.112.112.116
root@monitor1:~# snmptranslate -On NET-SNMP-EXTEND-
MIB::nsExtendOutputFull.\"battery\"
.1.3.6.1.4.1.8072.1.3.2.3.1.2.7.98.97.116.116.101.114.121
```

У вас эти номера могут отличаться!

Вписываем эти расширения (без точки перед единицей) в свой агент на другом компьютере и опрашиваем переменные. Сразу получаем строки по всем параметрам устройств в JSON формате.



Возможен иной вариант.

В локальных скриптах сразу извлекать в строковые переменные нужные данные по работе устройств.

А затем передавать переменным SNMP.

Пример работы с разделяемыми сегментами в net_map.php или net_mppt.php

Для работы через shell скрипт можно использовать обращение к БД Mysql (пример для МАП)

```
#!/bin/bash
out=`mysql -NBu root -pmicroart map -e "SELECT * from data WHERE
number=(SELECT MAX(number) from data)"`
```

Каждая переменная массива `${out[i]}` будет содержать соответствующее поле последней, добавленной в БД, строки.

Номера полей можно посмотреть через `/phpmyadmin`

(учетные данные: root, microart)

Имена переменных в основном соответствуют описанным в протоколе «МикроАрт»