Setup Olimex MICRO board with OliExt

1 – Network

Edit file `/etc/network/interface`

```
auto lo
iface lo inet loopback

auto eth0
#iface eth0 inet dhcp
iface eth0 inet static
address 192.168.1.50
netmask 255.255.255.0
gateway 192.168.1.1

#auto wlan1
#iface wlan1 inet dhcp
#wpa-ssid YourSSID
#wpa-psk YourPASS
```

DNS settings :

```
nano /etc/resolv.conf
```

2 – Use full SD Card space

```
resize_sd.sh /dev/mmcblk0 2
```

3 – Keyboard and locale

```
dpkg-reconfigure tzdata
dpkg-reconfigure locales
```

4 – Add user

```
adduser denis
```

5 – Update and Upgrade

```
apt-get update
apt-get upgrade
apt-get upgrade --fix-missing
```

6 – Colored bash

Edit file .bashrc and uncomment lines

```
export LS_OPTIONS='--color=auto'
eval "`dircolors`"
alias ls='ls $LS_OPTIONS'
alias ll='ls $LS_OPTIONS -l'
alias l='ls $LS_OPTIONS -lA'
```

7 – Install I2C support

```
apt-get install libi2c-dev
```

8 – Install GCC

```
apt-get install build-essential
```

9 – Install WiringOli

Copy file : `C:\Temp\Olinuxino\wiringOli\wiringOli` to `/home/denis`
Then install it with :

```
cd /home/denis/wiringOli/
make install
make install-static
```

You can compile example :

```
cd /home/denis/examples/
make really-all
```

10 – Changing FEX

Copy
`C:\Temp\Olinuxino\wiringOli\software\fex\Script.bin`
to
`/boot`
Then reboot.

11 - MySQL

Install MySQL with :

```
apt-get install mysql-server mysql-client
apt-get install libmysqlclient15-dev mysql-common
```

check MySQL is running :

```
mysqladmin -u root -p status
```

Edit `/etc/mysql/my.cnf`

Should be like :

```
#
# The MySQL database server configuration file.
#
# You can copy this to one of:
# - "/etc/mysql/my.cnf" to set global options,
# - "~/.my.cnf" to set user-specific options.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
```

```
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# This will be passed to all mysql clients
# It has been reported that passwords should be enclosed with ticks/quotes
# escpecially if they contain "#" chars...
# Remember to edit /etc/mysql/debian.cnf when changing the socket location.
[client]
port         = 3306
socket               = /var/run/mysqld/mysqld.sock

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

# This was formally known as [safe_mysqld]. Both versions are currently parsed.
[mysqld_safe]
socket               = /var/run/mysqld/mysqld.sock
nice         = 0

[mysqld]
#
# * Basic Settings
#
user         = mysql
pid-file     = /var/run/mysqld/mysqld.pid
socket               = /var/run/mysqld/mysqld.sock
port         = 3306
basedir              = /usr
datadir              = /var/lib/mysql
tmpdir               = /tmp
lc-messages-dir      = /usr/share/mysql
skip-external-locking
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address         = 127.0.0.1
#
# * Fine Tuning
#
key_buffer           = 32M
max_allowed_packet   = 16M
thread_stack         = 192K
thread_cache_size       = 8
# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover         = BACKUP
#max_connections        = 100
#table_cache            = 64
#thread_concurrency     = 10
#
# * Query Cache Configuration
#
query_cache_limit    = 2M
query_cache_size        = 32M
#
# * Logging and Replication
#
# Both location gets rotated by the cronjob.
# Be aware that this log type is a performance killer.
# As of 5.1 you can enable the log at runtime!
```

```
#general_log_file        = /var/log/mysql/mysql.log
#general_log             = 1
#
# Error log - should be very few entries.
#
log_error = /var/log/mysql/error.log
#
# Here you can see queries with especially long duration
#slow_query_log_file = /var/log/mysql/mysql-slow.log
#slow_query_log      = 1
#long_query_time = 2
#log_queries_not_using_indexes
#
# The following can be used as easy to replay backup logs or for replication.
# note: if you are setting up a replication slave, see README.Debian about
#       other settings you may need to change.
#server-id            = 1
#log_bin                      = /var/log/mysql/mysql-bin.log
expire_logs_days      = 10
max_binlog_size         = 100M
#binlog_do_db         = include_database_name
#binlog_ignore_db     = include_database_name
#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in /var/lib/mysql/.
# Read the manual for more InnoDB related options. There are many!
#
# * Security Features
#
# Read the manual, too, if you want chroot!
# chroot = /var/lib/mysql/
#
# For generating SSL certificates I recommend the OpenSSL GUI "tinyca".
#
# ssl-ca=/etc/mysql/cacert.pem
# ssl-cert=/etc/mysql/server-cert.pem
# ssl-key=/etc/mysql/server-key.pem



[mysqldump]
quick
quote-names
max_allowed_packet    = 16M

[mysql]
#no-auto-rehash       # faster start of mysql but no tab completition

[isamchk]
key_buffer            = 16M

#
# * IMPORTANT: Additional settings that can override those from this file!
#   The files must end with '.cnf', otherwise they'll be ignored.
#
!includedir /etc/mysql/conf.d/
```

Reboot.

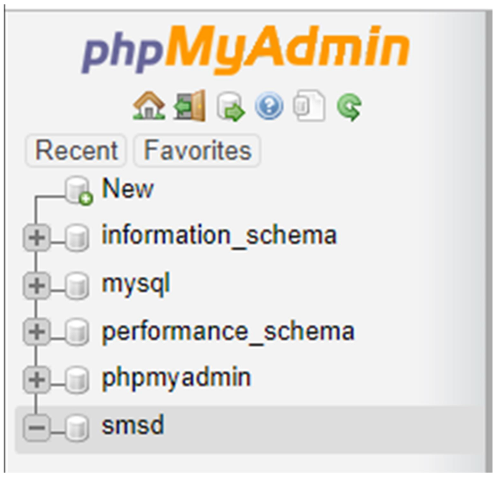
Then install phpmyadmin :

```
apt-get install phpMyAdmin
```

Choose apache2 as server.

Go to phpmyadmin : http://<IP Address>/phpmyadmin

Create a smsd database :



Get file mysql.sql file (install gammu first) :

```
cd /usr/share/doc/gammu/examples/sql
gunzip mysql.sql.gz
```

With phpmyadmin, run mysql.sql on the smsd database. This will create all tables :

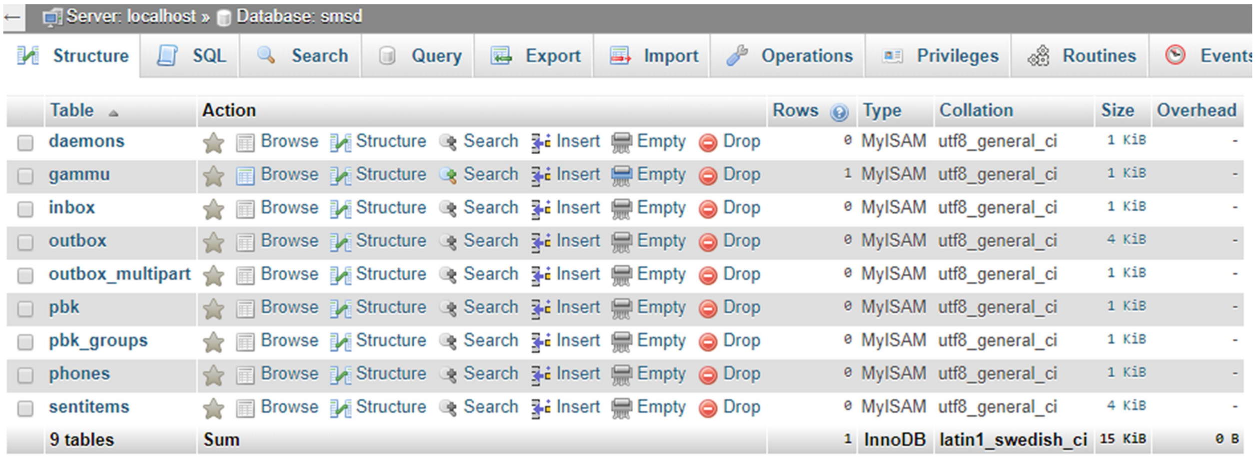


	Table	Action	Rows	Type	Collation	Size	Overhead
□	daemons	☆ Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1 KiB	-
□	gammu	☆ Browse Structure Search Insert Empty Drop	1	MyISAM	utf8_general_ci	1 KiB	-
□	inbox	☆ Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1 KiB	-
□	outbox	☆ Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	4 KiB	-
□	outbox_multipart	☆ Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1 KiB	-
□	pbk	☆ Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1 KiB	-
□	pbk_groups	☆ Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1 KiB	-
□	phones	☆ Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	1 KiB	-
□	sentitems	☆ Browse Structure Search Insert Empty Drop	0	MyISAM	utf8_general_ci	4 KiB	-
	9 tables	Sum	1	InnoDB	latin1_swedish_ci	15 KiB	0 B

12 – Gammu

```
apt-get install gammu
apt-get install gammu-smsd
gammu-detect /etc/gammurc
nano /etc/gammurc
```

Set gammu configuration files /etc/gammurc :

```
# This is a generated gammurc file.
# It was generated by Gammu configurator 0.4

# In Unix/Linux  : copy it into your home directory and name it .gammurc
#                  or into /etc and name it gammurc
# In Win32       : copy it into directory with Gammu.exe and name gammurc

# Port           : in Windows/DOS: "com*:",
#                  (instead of "*" please put "1", "2", etc.)
#                  in other (Linux/Unix) "/dev/ttyS%"
#                                   or "/dev/ircomm%" ("irda" connection)
#                  (instead of "%" please put "0", "1", "2", etc.)
# Model          : use only, when Gammu doesn't recognize your phone model.
#                  Put it here. Example values: "6110", "6150", "6210", "8210"
# Connection     : type of connection. Use "fbus" or "mbus" or "dlr3" or
#                  "irda" (Infrared over sockets) or "infrared" (DirectIR)
#                  or "at19200" (AT commands on 19200, 8 bits, None parity,
#                  1 stop bit, no flow control) or "at115200" (AT commands on
#                  115200, 8 bits, None parity, 1 stop bit, no flow control)
#                  or "atblue" (AT over BlueTooth) or "dlr3blue" (FBUS
#                  over BlueTooth)
# SynchronizeTime: if you want to set time from computer to phone during
#                  starting connection. Do not rather use this option when want
#                  to reset phone during connection (in some phones need to
#                  set time again after restart)
# Logfile        : Use, when want to have logfile from communication.
# Logformat      : What debug info and format should be used:
#                  "nothing" - no debug level, "text" - transmission dump in
#                  text format, "textall" - all possible info in text format,
#                  "errors"  - errors in text format, "binary" - transmission
#                  dump in binary format
# Use_Locking    : under Unix/Linux use "yes", if want to lock used device
#                  to prevent using it by other applications
# GammuLoc       : name of localisation file

[gammu]

port = /dev/ttyS4
model = at
connection = at9600
synchronizetime = yes
logfile =
logformat = nothing
use_locking =
gammuloc =
name=SIM900
```

Set gammu configuration files /etc/gammu-smsdrc :

```
# Configuration file for Gammu SMS Daemon

# Gammu library configuration, see gammurc(5)
[gammu]
port = /dev/ttyS4
connection = at9600
```

```
# Debugging
#logformat = textall

# SMSD configuration, see gammu-smsdrc(5)
[smsd]
service = SQL
driver = native_mysql
database = smsd
user = root
password = DMO12lip6
pc = localhost
logfile = /var/log/gammu.log
phoneid = SIM900
CheckSecurity = 0
# Increase for debugging information
debuglevel = 1
loopSleep = 10
StatusFrequency = 60
ReceiveFrequency = 60
CommTimeout = 60
SendTimeout = 60

# Paths where messages are stored
# inboxpath = /var/spool/gammu/inbox/
# outboxpath = /var/spool/gammu/outbox/
# sentsmspath = /var/spool/gammu/sent/
# errorsmspath = /var/spool/gammu/error/
```

start gammu with :

```
gammu-smsd --daemon
```

Look if everything is working fine in the log file :

```
/var/log/gammu.log
```

13 - Programs

Copy your programs in `/home/denis` and you can run them

To be able to compile shutter run this commands first :

```
mysql_config --include
mysql_config --libs
mysql_config –cflags
apt-get install libncurses5-dev
```

Compile programs with :

```
gcc -Wall -o PowerOnGSM PowerOnGSM.c –lwiringOli

gcc -Wall -o shutter shutter4.c teleinfo.c -lwiringOli $(mysql_config --cflags)
$(mysql_config --libs)

gcc -Wall -o sms main.cpp -lwiringOli $(mysql_config --cflags) $(mysql_config --
libs)
```

14 – Debug

Install GDB :

```
apt-get install gdb
```

Then compile with –g :

```
gcc -Wall -o sms main.cpp -lwiringOli $(mysql_config --cflags) $(mysql_config --libs) -g
```

Run program with GDB :

```
gdb sms
```

at the GDB prompt (gdb) just type r to run.

15 – Screen

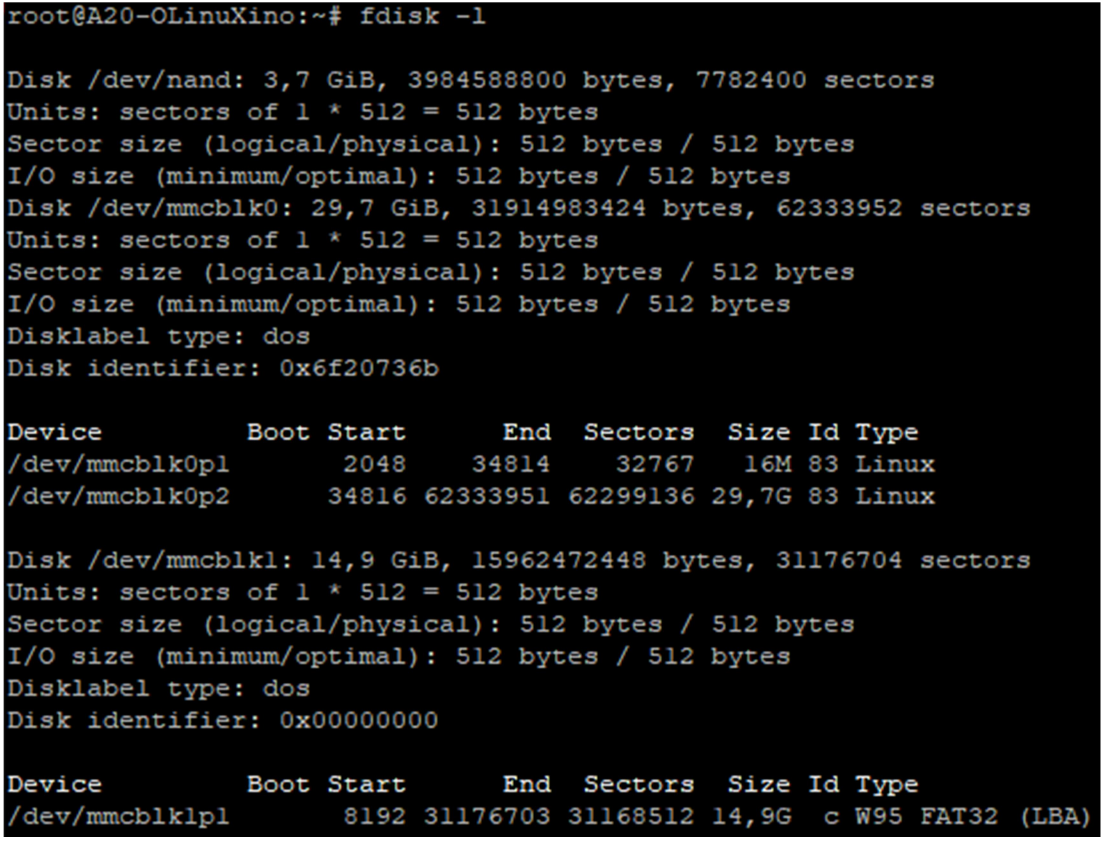
```
sudo apt-get install screen
```

16 – Clone SD Card

Insert SD Card to backup in the SD card port of Olimex board (Not on the micro SD port)
Boot on the micro SD

Do a `fdisk -l` to see the list of disk

You see something like :



```
root@A20-OLinuXino:~# fdisk -l

Disk /dev/nand: 3,7 GiB, 3984588800 bytes, 7782400 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk /dev/mmcblk0: 29,7 GiB, 31914983424 bytes, 62333952 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x6f20736b

Device          Boot Start      End  Sectors  Size Id Type
/dev/mmcblk0p1        2048    34814    32767   16M 83 Linux
/dev/mmcblk0p2       34816 62333951 62299136 29,7G 83 Linux

Disk /dev/mmcblk1: 14,9 GiB, 15962472448 bytes, 31176704 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device          Boot Start      End  Sectors  Size Id Type
/dev/mmcblk1p1        8192 31176703 31168512 14,9G  c W95 FAT32 (LBA)
```

Then run `dd if=/dev/mmcblk1 of=myImage.img`

To copy the image to a new SD Card, put the new SD card in the SD Card reader (Not the micro SD port).

Then run :

```
dd if=myImage.img of=/dev/mmcblk1
```