

## MOUSE2Go Quickstart Anleitung Raspberry Pi Version

Diese Anleitung beschreibt die einzelnen Schritte um MOUSE2Go auf einem frisch installierten Raspberry Pi in Verbindung mit einem Arduino zum Laufen zu bringen.

- 1) Herunterladen und Installieren der aktuellen raspbian-jessie GUI Version

(<https://www.raspberrypi.org/downloads/raspbian/>)

Nach dem initialen Einrichten sollte der User "pi" auf dem grafischen Oberfläche Pixel eingeloggt sein.

- 2) Als nächstes muss die aktuelle Arduino IDE installiert werden. (<https://www.arduino.cc/en/Main/Software>)  
Download der Linux-ARM Version

Öffnen eines Terminals und ausführen der folgenden Befehle:

```
cd ~/Downloads
tar -xJf arduino-1.8.1-linux-arm.tar.xz
cd arduino-1.8.1
./installer.sh
```

Damit sollte auf dem Desktop ein Icon zum Starten der Arduino-IDE auftauchen. Ggf. muss die Versionsnummer (1.8.1) in den Befehlen der tatsächlichen Nummer in der Downloaddatei angepasst werden.

Starten der IDE und testweise übertragen eines Programms auf den per USB verbundenen Arduino.

Detaillierte Hilfe gibt es hier: <https://www.arduino.cc/en/Guide/HomePage>

- 3) Als nächstes müssen einige zusätzliche Tools installiert werden. Öffnen eines Terminals und ausführen der folgenden Befehle:

```
sudo apt-get install -y screen git cutecom
```

- 4) Nun laden wir die aktuelle Version von MOUSE2Go herunter, dafür wieder im Terminal folgendes ausführen:

```
cd ~/Documents
git clone https://github.com/mkeller0815/MOUSE2Go
```

- 5) In der Arduino IDE öffnen wir nun den Sketch im Verzeichnis

/home/pi/Documents/MOUSE2Go/Arduino/arduino\_6502\_mouse/ arduino\_6502\_mouse.ino und laden ihn auf den Arduino hoch. Wer einen Arduino Mega oder Arduino Due verwendet, der kann in die folgende Zeilen im Sketch ändern:

```
// set this to a proper value for your Arduino
#define RAM_SIZE 1536 // Uno, mini or any other 328P
// #define RAM_SIZE 6144 //Mega 2560
// #define RAM_SIZE 32768 //Due
```

zu

```
// set this to a proper value for your Arduino
// #define RAM_SIZE 1536 // Uno, mini or any other 328P
#define RAM_SIZE 6144 //Mega 2560
// #define RAM_SIZE 32768 //Due
```

Damit steht dem 6502 System mehr RAM zur Verfügung.

- 6) Nun können wir das erste Mal mit MOUSE interagieren. Dazu starten wir in einem Terminal das Tool „screen“. Zusätzlich muss auch die serielle Schnittstelle angegeben werden. Hier verwenden wir die gleiche wie die Arduino IDE. z.B. /dev/ttyACM0

```
screen /dev/ttyACM0 9600
```

Das Kommando „h“ liefert einen Überblick über die Befehle des Monitors. Ein Reset des Arduino, setzt das 6502 System wieder zurück und mittels „<CTRL>-<a>“ gefolgt von einem „k“ kann man das Terminal wieder verlassen. Das ist auch notwendig, da sonst die serielle Schnittstelle blockiert ist.

7) Download Ophis Assembler unter

<https://github.com/michaelcmartin/Ophis/releases/download/v2.1/Ophis-2.1.tgz>

und Installieren mit:

```
cd ~/Downloads
tar -xzf Ophis-2.1.tgz
cd Ophis-2.1/
sudo mv Ophis-2.1 /opt/
sudo ln -s /opt/Ophis-2.1/ophis /usr/local/bin/
```

Somit kann der Assembler überall aufgerufen werden

8) Hello-World in 6502 Assembler:

Anlegen einer Datei „hello.asm“ mit folgendem Inhalt:

```
.org      $0400                      ; start of programm

loop:     ldx #$00                    ; set index
          lda hello,x                ; load character
          beq done                    ; 0 means done
          jsr $e003                  ; print character
          inx                        ; increment index
          bne loop                   ; next character
done:     jmp ($fffc)                 ; reset to monitor
hello:    .byte 13,"HELLO, WORLD!",13, 0
```

Assemblieren mit „ophis -v -o hello.bin hello.asm“

Zur Zeit gibt es noch keine einfache Möglichkeit das Programm zu übertragen, daher greifen wir zu einem Trick.

Umwandeln der binären Daten in HEX-Werte:

```
hexdump -e '16/1 "%02x" "\n"' hello.bin
```

Das ergibt folgende Ausgabe:

```
a200bd1004f0062003e0e8d0f56cfcff
0d48454c4c4f2c20574f524c44210d00
```

Diesen Block von Hex-Werten können wir nun im MOUSE Terminal mit dem Befehl

```
i 0400
```

importieren. Dazu die Zeilen einzeln kopieren und im Terminal wieder einfügen.

Am Ende einen einzelnen „.“ eingeben, um die Eingabe abzuschliessen. Ein

```
m 0400 10 02
```

sollte dann die Werte wieder ausgeben.

Mittels

```
g 0400
```

kann das Programm ab Adresse 0x0400 ausgeführt werden. Das „jmp (\$fffc)“ am Ende des Programms sorgt für einen Reset und damit einen Sprung in das Monitorprogramm.