Setting up TSC - alpha with a Raspberry Pi 3 Model B v. 0.0.1

TSC (TwoStepperControl) is a software project by Wolfgang Birkfellner and Steffen Elste to realize n open integrated control system for astronomical telescopes. The basic idea is to use a Raspberry Pi to control a telescope and to provide basic autoguiding functionality without additional external computers. As a Pi is not a RTOS, it cannot control stepper drivers directly like a microcontroller; therefore, extra hardware is necessary which receives basic parameters (number of steps, velocity, acceleration) from the software. Currently, I am using a Phidgets 1067 bipolar driver board although other alternatives are under consideration. The steppers are monitored by the Pi in separate Qt-Concurrent threads so that software functionality is maintained during stepper operation. So far, only fork mounts are supported.

Software repository:
https://github.com/selste/TwoStepperControl

Basic components:

- Operating System: Raspian Jessie - https://www.raspberrypi.org/downloads/noobs/
- Raspberry Pi 3 Model B, with at least a 16 GB MicroSD card and a 2A micro-USB power adapter. Cost ~ 70 €.
- Stepper control boards: 1067_0 - PhidgetStepper Bipolar HC, available from www.phidgets.com or www.nodna.de. Two boards are necessary. These operate one bipolar stepper each with up to 4 A coil current. 1/16 microsteps are available, the board is controlled via USB 2.0. Cost for each board is currently ~90 €. A development with the SparkFun AutoDriver - Stepper Motor Driver (v13) – www.sparkfun.com - using STMicro L6470 dSPIN drivers capable of 3A per coil and 1/128 microsteps addressed via SPI is planned but not yet available. As Raspian is not a RTOS, it cannot control the drivers directly.
- A HDMI cable and an HDMI monitor. Development right now aims at using the Adafruit 5” HDMI 800*480 Backpack Touchscreen (~80 €). Basically, it is possible to connect via W-LAN or WI-FI direct and a VNC viewer to the Raspberry using a mobile or a tablet, although I have not yet tried this – check https://www.raspberrypi.org/documentation/remote-access/vnc/
- A wireless mouse and keyboard. Cost ~20 €.
- I am experimenting right now with establishing a solid RS232 connection to connect to Cartes du Ciel. For this purpose, it is necessary to get a simple level shifter circuit like this one: https://www.conrad.at/de/rs232-erweiterungs-platine-fuer-den-raspberry-pi-1337093.html However, if you don't have a serial port and a NullModem – cable, it might make sense to use one of these although I have not tried it yet: https://www.adafruit.com/product/954 . Finding out about available RS232 port can be done in UNIX-type systems by typing
  `dmesg | grep tty`
- A Raspberry Pi compatible USB 2.0 hub with separate power. Not all hubs work, so be careful. I found the USB port in the lower left of the Pi to be the most stable.

- Development environment is Qt and QtCreator using C++. Be aware that some versions of Jessie might give trouble when starting the QtCreator – try
  `qtcreator -noload Welcome` and give the correct paths to compilers and debuggers in the Options – menu.

- **Software requirements:**
  - Install the Phidgets – library and set the right access privileges for the USB port if you don't want to run the program in `root` mode. Check http://www.instructables.com/id/Getting-Started-with-Phidgets-on-the-Raspberry-Pi/ … Basically, you need the libusb 1.0 and libphidgets 2.1. Changing the udev rules is also necessary (this is taken from http://tordwessman.blogspot.co.at/2012/01/running-phidgets-under-linuxubuntu.html):
    - `$ sudo nano /etc/udev/rules.d/80_phidget.rules`
    - Add the following content:
      `SUBSYSTEMS=="usb", ACTION=="add", ATTRS{idVendor}=="06c2", ATTRS{idProduct}=="00[3-a][0-f]", MODE="666"`
    - Set `USB_DEVFS_PATH` to `/dev/bus/usb` by adding to `~/.bashrc`:
      `export USB_DEVFS_PATH=/dev/bus/usb`
    - Restart udev:
      `$ services udev restart`
    - *I*f it doesn't work. Try the good old:
      `$ sudo reboot`
  - Currently, only image acquisition of a QHY5 CCD-camera is realized via INDI; other cameras may follow. However, one needs INDI, and a tutorial for installing it on the Pi is found under
    http://indilib.org/support/tutorials/139-indi-library-on-raspberry-pi.html
  - In order to utilize the serial port, one has to disable it's current purpose – serving as a console to the pi.
    - Edit `/boot/config.txt` by typing `enable_uart=1.`
    - Disable console output from Jessie by typing:
      `$ sudo systemctl stop serial-getty@ttyS0.service`
      `$ sudo systemctl disable serial-getty@ttyS0.service`
    - Type sudo nano /boot/cmdline.txt and remove the statement `console=serial0,115200` from that text. After reboot, `/dev/ttyS0` should be available. More can be found under

      http://spellfoundry.com/2016/05/29/configuring-gpio-serial-port-raspbian-jessie-including-pi-3/

Current features:

- Independent control of two Phidget steppers. Mount and drive parameters can be directly edited and saved in the GUI of TSC.
- Motion of the mount at variable speed using a handbox-style interface.
- GoTo using an internal set of catalogues; the catalogues are named `.tsc` but are in fact `.csv` files that can be edited with every spreadsheet like Excel. They have the following structure
  - Number of entries in the first cell

    - Constellation, Object name, RA hour, RA minute, RA second, sign of the declination, declination degrees, declination minutes, declination seconds.
    - Example (start of the Messier list):

      ```
      110,,,,,,,,
      Taurus,Messier 1,5,34,30,+,22,1,0
      Aquarius,Messier 2,21,33,30,-,0,49,0
      ```

      and so on ...

- Interfacing to a local INDI server for ccd-camera acquisition; this is still very, very basic
- Acquisition of images from a guiding camera – this is even more basic. Currently, only the QHY5 camera is supported.
- Serial interface to a LX200 basic client; this is really, really bleeding ...