RoboCar

Building a minimalistic robot

Using TCL as controlling software

Gerhard Reithofer
2022-06-30, 2022-07-01
EuroTCL 2022, Vienna
Components

- Control unit: Raspberry Pi 1B (2012) + Pi camera module
- 2 stepper motors (28BYJ-48) + driver (ULNA2003)
- Powered by an Accu pack (Intenso PM5200)
- 3D printed mechanical parts, cad models and STL files
- Control network: Wireless USB-Lan, Pi 3+ have wlan
- TCL Software for controlling Stepper motors
- Images/photos are managed over VLC-client/server
- The control network is SSH a connection in a wlan
Raspberry Pi 1B (2012)

https://www.mbtechworks.com/hardware/raspberry-pi-model-comparison.html

Camera module: https://www.reichelt.at/
## Raspberry Pi 1B (2012)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Pi 1B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Released</td>
<td>Apr 2012</td>
</tr>
<tr>
<td>Architecture</td>
<td>ARMv6Z, 32-bit</td>
</tr>
<tr>
<td>SoC</td>
<td>Broadcom BCM2835</td>
</tr>
<tr>
<td>CPU</td>
<td>700 MHz, ARM1176JZF-S</td>
</tr>
<tr>
<td>Cores</td>
<td>1</td>
</tr>
<tr>
<td>GPU</td>
<td>Broadcom VideoCore IV HD 1080p</td>
</tr>
<tr>
<td>Memory RAM</td>
<td>512 MB</td>
</tr>
<tr>
<td>Operating System</td>
<td>Primarily Linux based</td>
</tr>
<tr>
<td>USB 2.0 Ports</td>
<td>2</td>
</tr>
<tr>
<td>Camera Input</td>
<td>15-pin CSI (Camera Serial Interface)</td>
</tr>
<tr>
<td>Video Output</td>
<td>Composite 3.5 mm RCA and HDMI</td>
</tr>
<tr>
<td>Audio Output</td>
<td>Analog 3.5 mm jack, Digital via HDMI</td>
</tr>
<tr>
<td>Storage</td>
<td>SD slot</td>
</tr>
<tr>
<td>Ethernet</td>
<td>10/100 Mbps</td>
</tr>
<tr>
<td>Input/Output Pins</td>
<td>26</td>
</tr>
<tr>
<td>Power (less peripherals)</td>
<td>5v 700 ma</td>
</tr>
<tr>
<td>Size</td>
<td>85 mm x 56 mm</td>
</tr>
</tbody>
</table>

- USB-wlan Stick
- No bluetooth
- No wlan on board

Controlling is done via SSH remote session
Stepper motor movement


Controlled by single repeatable steps

4 signals are sufficient for reliable movement
Power Hardware

Powerbank
Intenso PM5200

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charging slots</td>
<td>1</td>
</tr>
<tr>
<td>Capacity</td>
<td>5200 mAh</td>
</tr>
<tr>
<td>Charging current (max.)</td>
<td>1000 mA</td>
</tr>
<tr>
<td>Technology</td>
<td>Li-ion</td>
</tr>
<tr>
<td>Weight</td>
<td>136 g</td>
</tr>
<tr>
<td>Dim (L x W x H)</td>
<td>96 x 43 x 25 mm</td>
</tr>
<tr>
<td>Connector type</td>
<td>USB-A 1x, Micro USB 1x</td>
</tr>
<tr>
<td>Powerbank features</td>
<td>Status display</td>
</tr>
<tr>
<td>USB-C current input (max.)</td>
<td>1 A</td>
</tr>
<tr>
<td>USB-C voltage input (max.)</td>
<td>5 V</td>
</tr>
<tr>
<td>USB-C current output (max.)</td>
<td>1 A</td>
</tr>
<tr>
<td>USB-C voltage output (max.)</td>
<td>5 V</td>
</tr>
<tr>
<td>Product type</td>
<td>Power bank</td>
</tr>
</tbody>
</table>
CAD design + 3D printing
Hardware/Wiring

Physical connections

G eneral
P urpose
I nput
O utput

Simple cheap jumper cables
Software/Hardware Interface

Communication via /sys/class/gpio filesystem.

GPIO Directory: export unexport gpiochip0

```
set fd [open /sys/class/gpio/export {WRONLY}]; puts $fd "15"; close $fd
After export: export unexport gpiochip0 gpio15
```

GPIO15 directory: direction active_low device subsystem edge power uevent value
Setting direction to 'out'
```
set fd [open /sys/class/gpio/gpio15/direction {WRONLY}]; puts $fd "out"; close $fd
```

Writing value "1" to port 15
```
set fd [open /sys/class/gpio/gpio15/value {WRONLY}]; puts $fd "1"; close $fd
```

Destroying connection to port 15
```
set fd [open /sys/class/gpio/unexport {WRONLY}]; puts $fd "15"; close $fd
```

GPIO Directory: export unexport gpiochip0

Module: raspi2.tcl
Used Tools

Remote Shell to start a control session

```
ssh -l pi -X robocar
```

Video/Photo connection:
VLC as client and server for image/video transfer

```
$ ./live_video.sh
Call: vlc tcp/h264://192.168.0.192:3333
on remote machine to view.
Waiting for a TCP connection on 0.0.0.0:3333...
```
Programming

Motion control: Tcl 8.6
Command line based robot controlling
No specific extension is required

Usage: ULNA2003.tcl [-h] [-a {axid1 [axid2]}] [-d delay] [-t drivetype] [-x macro] [-f {0|1}] wheelsteps
Default axis ids are: st0..st5
-h .. command line syntax help
-a .. stepper motor id (axis) st0..st5 (Def. st2 st0)
-d .. stepper delay value (min=1) (def. 1)
-f .. fix end position (def. 0)
-t .. drive id - one of f, d, m, h (def. d)
-x .. execute macro
The wheelsteps value must be a sequence of steps for L/R wheel
Possible cycle ids are:
   f - full step mode - 4 micro steps
   d - drive step mode - 4 micro steps
   m - mix of full and halve - 6 micro steps
   h - halve step mode - 8 micro steps
See also: https://dronebotworkshop.com/stepper-motors-with-arduino/
Final Demonstration

Thank you for your attention