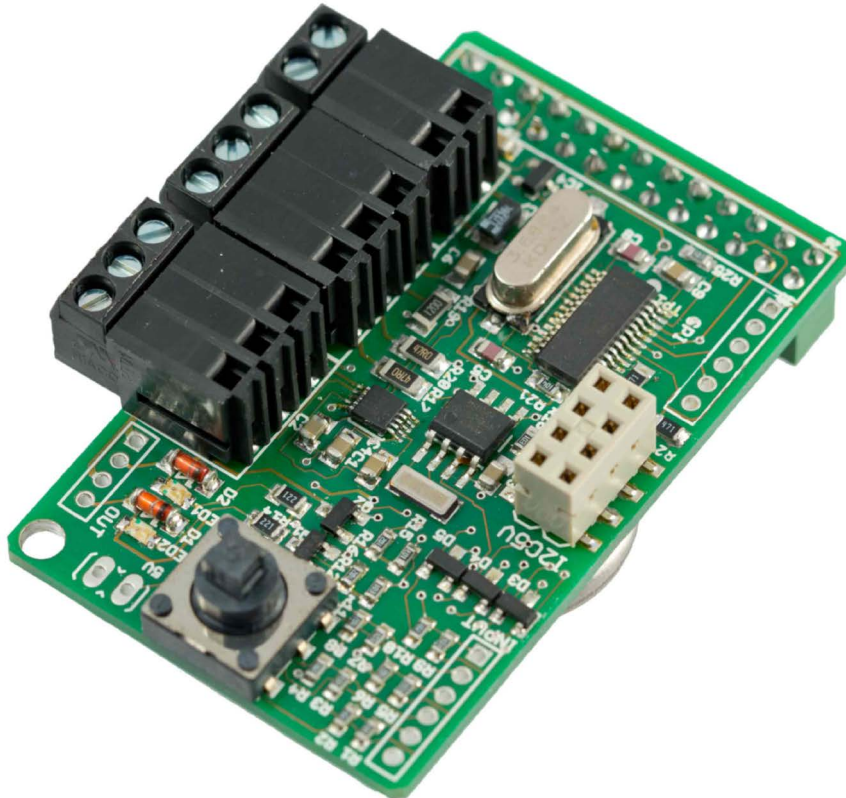


Extension board for the Raspberry Pi single-board computer



Main Features

- **RS-485 port** – for control over stepper motors, etc.
- **RS-232 port** – connect to serial port devices like a modem or your PC
- **I2C connector** – directly connect a display or sensors
- **real-time clock with battery backup**
- **5-way joystick**
- piggyback mounting – no cables or additional space required
- driver support and sample applications code
- fully assembled including the backup-battery

Table of Contents

1. Technical Specifications.....	2
2. Technical Drawing	3
3. First Time Installation	4
3.1. Choice of distribution	4
3.2. Find out your Raspberry Pi's revision	4
3.3. Raspicomm Setup	5
3.4. Setup-Script Parameters	5
4. Using your Raspicomm	5
4.1. RS-485 Port	5
4.2. Serial Port / RS-232.....	6
4.3. Joystick	6
4.4. Outputs (LEDs).....	6
4.5. HWClock	7
5. Additional Links	7

1. Technical Specifications

Size: 35,2 x 56mm

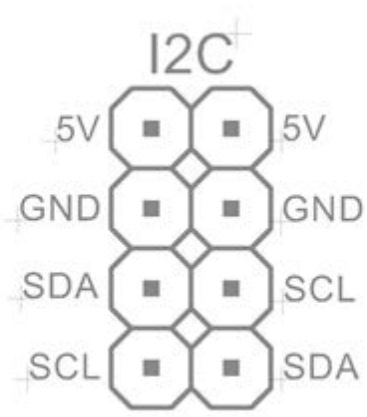
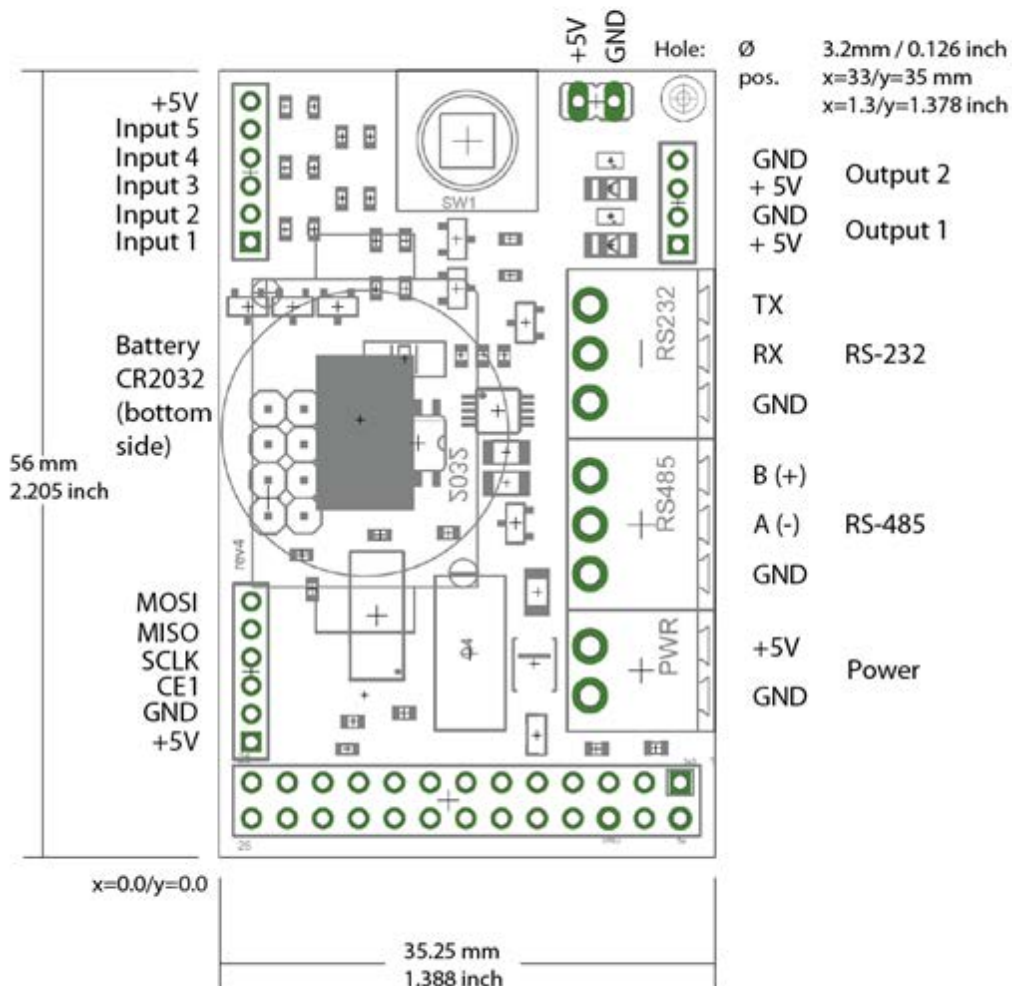
- **RS-485**
 - up to 230.400 baud
 - pluggable screw terminal connector (A, B, GND)
- **RS-232**
 - No hardware handshake
 - up to 115.200 baud
 - pluggable screw terminal connector (RX, TX, GND)
- **Real Time Clock**
 - CR2032 RTC backup battery included
 - powers the clock for approximately 10 years
- **5 Inputs**
 - connected to onboard joystick (4 way + push)
 - 5V tolerant
 - 2mm pitch external connector, not populated
- **2 Outputs**
 - 5V, 100mA max. each
 - 5V relays can be connected without additional circuitry due to onboard protection diodes
 - 2mm pitch connector, not populated
- **SPI Connector**
 - 2mm pitch connector, not populated
- **2 I2C Connectors**
- **Power Connector**
 - pluggable screw terminal connector (+, -)
 - either powers the Raspberry Pi and RasPiComm (5V, 1.5A max)
 - or is a power output when powered over USB
- **RasPiComm Power Consumption**
 - max . 10mA (outputs off)
 - max. 210mA (outputs on and maximum rating drawn)



2. Technical Drawing

RASPiCOMM

TECHNICAL DRAWING



3. First Time Installation

3.1. Choice of distribution

Install a distribution that supports *i2c* and the *spi bus*.

1. Download Distribution:
 - either [Raspbian 3.6.11+ Kernel #538 \(2013-09-10\)](#)
 - or [Raspbian 3.6.11+ \(2013-07-26\)](#)
 - or [Raspbian 3.6.11+ \(2013-05-25\)](#)
 - or [Raspbian 3.6.11+ \(2013-02-09\)](#)
 - or [Raspbian 3.2.27+ \(2012-12-15\)](#)

2. Save it to the SD card

You can find further instructions for setting up your SD-Card [here](#).

3. Boot Raspberry Pi from the SD card

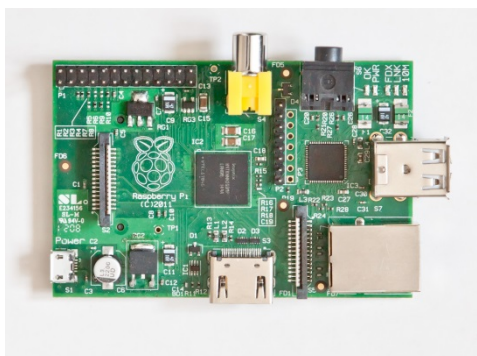
4. Configure your distribution on first boot

To reconfigure your distribution later you can always use the following command:

```
raspi-config
```

3.2. Find out your Raspberry Pi's revision

Revision 1 looks like this:



Raspberry Pi rev1 (Source: wikimedia.org)

Revision 2 can be identified by the mounting hole on the top right corner.



Raspberry Pi rev2 (Source: raspberrypi.org)

3.3. Raspicomm Setup

Execute the setup-script from your home directory (cd ~)

```
wget http://downloads.amescon.com/rpc_setup.sh && chmod +x ./rpc_setup.sh && sudo ./rpc_setup.sh
```

The setup-script requires the following actions:

- Choose revision according to your Raspberry Pi (switch revisions with the 'c'-button)
- Activate installation script (with the 'i'-button)
 - This script will install i2c-tools and permanently set up the following devices:
 - Joystick
 - RS-485
 - RS-232
 - Output (LEDs)
 - HWClock
- Confirm driver installation with 'y'.
- For the RS-232 to work, you'll need to restart your Raspberry Pi. All other devices are ready to go without a restart.

3.4. Setup-Script Parameters

The script can be run with parameters that are however not needed for a routine setup. For a full list of parameters please use one of the following commands:

```
./rpc_setup.sh /?
./rpc_setup.sh --h
./rpc_setup.sh --help
```

While the help-parameters do not need root access, most others will. For example, the superuser-elevated command

```
sudo ./rpc.setup.sh --remove-autostart
```

will remove the autostart-sequence for your Raspicomm.

4. Using your Raspicomm

4.1. RS-485 Port

After the device driver is installed, the RS-485 port is added under this name:

```
/dev/ttyRPC0
```

You can use linux standard libraries to access the port.

Example code for C or C++:

```
int fd = open("/dev/ttyRPC0", O_RDWR | O_NOCTTY | O_NDELAY); ...
```

More code examples can be found here: [Sample Applications](#).



4.2. Serial Port / RS-232

By default the RS-232 port is used by the Raspberry Pi's startup routine and needs to be reconfigured for use. The setup script does so automatically. Please note that a restart is required, after which you can use standard libraries to access the serial port. The device name is:

```
/dev/ttyAMA0
```

Example code for C or C++:

```
int fd = open("/dev/ttyAMA0", O_RDWR | O_NOCTTY | O_NDELAY); ...
```

Our [Sample Applications](#) can be adapted for use with the RS-232 port by replacing the RS-485 name */dev/ttyRPC0* by its RS-232 counterpart */dev/ttyAMA0*.

4.3. Joystick

The joystick is mapped to the following GPIOs:

1. 4 = right side (towards HDMI)
2. 22 = push
3. 23 = bottom side (towards power connection)
4. 24 = left side (towards audio out)
5. 25 = top side (towards usb)

You can use linux standard libraries to access the joystick in the same way as the serial port.

Example code for C or C++:

```
int fd = open("/sys/class/gpio/gpio22/value", O_RDONLY); ...
```

4.4. Outputs (LEDs)

The Output is mapped to the following GPIOs:

- Output 1 (D1): GPIO 18
- Output 2 (D2): GPIO 21 (Raspberry Pi rev.1) or GPIO 27 (Raspberry Pi rev.2)

4.5. HWClock

The setup-script will synchronize your system time with the HWClock every time your Raspberry Pi is restarted.

This command will synchronize your HWClock with the system time:

```
sudo hwclock --systohc
```

This command will manually synchronize your system time with the HWClock:

```
sudo hwclock --hctosys
```

To manually set your HWClock, use the following command:

```
sudo hwclock --set --date='mm/dd/yy hh:mm:ss' --localtime
```

5. Additional Links

Should you require further information or are looking for project code, check out the following links:

- [Raspicomm F.A.Q](#): Answers to the most common questions.
- [Amescon Github](#): Our open-source sample applications for your Raspicomm.
- [Raspicomm Support Forum](#): If the manual and F.A.Q could not answer your questions, this is the place to go.
- [Online-Version of this manual](#)

