# RASPBERRY PI I/O EXPANSION CARD
for BUILDING AUTOMATION

USER'S GUIDE VERSION 1.0
- preliminary-

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GENERAL DESCRIPTION</td>
<td>2</td>
</tr>
<tr>
<td>WHAT IS IN YOUR KIT</td>
<td>3</td>
</tr>
<tr>
<td>QUICK START-UP GUIDE</td>
<td>3</td>
</tr>
<tr>
<td>BOARD LAYOUT</td>
<td>4</td>
</tr>
<tr>
<td>RS-485/MODBUS COMMUNICATION</td>
<td>5</td>
</tr>
<tr>
<td>STACK LEVEL JUMPERS</td>
<td>6</td>
</tr>
<tr>
<td>RASPBERRY PI HEADER</td>
<td>7</td>
</tr>
<tr>
<td>POWER REQUIREMENTS</td>
<td>8</td>
</tr>
<tr>
<td>SCHEMATICS</td>
<td>9</td>
</tr>
<tr>
<td>HARDWARE WATCHDOG</td>
<td>11</td>
</tr>
<tr>
<td>ANALOG INPUTS/OUTPUTS CALIBRATION</td>
<td>12</td>
</tr>
<tr>
<td>HARDWARE SPECIFICATIONS</td>
<td>13</td>
</tr>
<tr>
<td>LINEARITY OVER FULL SCALE</td>
<td>14</td>
</tr>
<tr>
<td>MECHANICAL SPECIFICATIONS</td>
<td>15</td>
</tr>
<tr>
<td>SOFTWARE SETUP</td>
<td>16</td>
</tr>
<tr>
<td>SCHEMATICS</td>
<td>17</td>
</tr>
</tbody>
</table>
GENERAL DESCRIPTION

The second generation of our Building Automation card brings to the Raspberry Pi platform all the building blocks required for Building Automation projects. Stackable to 8 levels, the card works with all Raspberry Pi versions, from Zero to 4.

Two of the Raspberry Pi’s GPIO pins are used for I2C communication. Another pin is allocated for the interrupt handler, leaving 23 GPIO pins available for the user.

FEATURES at a glance:

- Eight jumper settable universal, analog/digital inputs
  - 0-10V Inputs or
  - Contact Closure Counter Inputs or
  - 1K/10K Temperature Sensor Inputs
- Four 0-10V Outputs
- Four TRIAC Outputs with 1A/48VAC drivers
- Four General Purpose LED’s
- RS485 in and out ports
- Real time clock with battery backup
- On-board push-button
- On-board Hardware Watchdog
- 24VAC/DC power supply

All inputs and output use pluggable connectors which permit easy wiring access when multiple cards are stacked.

The four general purpose LED’s can be associated with the analog inputs or other controlled processes.

The on-board push button can be programmed to cut inputs or override outputs.
WHAT IS IN YOUR KIT

1. MEGA-BAS add-on card for Raspberry Pi
2. Mounting hardware
   a. Four M2.5x18mm male-female nylon standoffs
   b. Four M2.5x5mm nylon screws
   c. Four M2.5 nylon nuts
3. Two jumpers. You do not need the jumpers when using only one MEGA-BAS card. See STACK LEVEL JUMPERS section on page 6 if you plan to use multiple MEGA-BAS cards.
4. Four 8-pin female mating connectors.

QUICK START-UP GUIDE

1. Plug your MEGA-BAS card on top of your Raspberry Pi and power up the system.
2. Enable I2C communication on Raspberry Pi using raspi-config.
3. Install the MEGA-BAS software from github.com:
   a. ~$ git clone https://github.com/SequentMicrosystems/megabas-rpi.git
   b. ~$ cd /home/pi/megabas-rpi
   c. ~/megabas-rpi$ sudo make install
4. ~/megabas-rpi$ megabas
   The program will respond with a list of available commands.
BOARD LAYOUT

EIGHT UNIVERSAL INPUTS

INPUT SELECT JUMPERS

485-IN

485-OUT

STACK LEVEL

485-EN

485-TX

485-RX

CR2032 Battery Holder

RESERVED

SEQUENTMICROSYSTEMS.COM MEGA-BAS REV-2.0

24VAC/DC

STATUS LEDS

GP LEDS

FOUR TRIAC OUTPUTS

FOUR 0-10V OUTPUTS

Raspberry Pi Connector
RS-485/MODBUS COMMUNICATION

The MEGA-BAS card contains a standard RS485 transceiver which can be accessed both by the local processor and by Raspberry Pi. The desired configuration is set from three bypass jumpers on configuration connector J3.

If jumpers are installed, Raspberry Pi can communicate with any device with an RS485 interface. In this configuration the MEGA-BAS card is a passive bridge which implements only the hardware levels required by the RS485 protocol. To use this configuration, you need to tell the local processor to release control of the RS485 bus:

```
~$ mega bas [0] wcfgmb 0 0 0 0
```

If jumpers are removed, the card operates as MODBUS slave and implements the MODBUS RTU protocol. Any MODBUS master can access all the card's inputs, and set all the outputs using standard MODBUS commands. A detailed list of commands implemented can be found on GitHub:

[https://github.com/SequentMicrosystems/megabas-rpi/blob/master/Modbus.md](https://github.com/SequentMicrosystems/megabas-rpi/blob/master/Modbus.md)

In both configurations the local processor needs to be programmed to release (jumpers installed) or control (jumpers removed) the RS485 signals. See the command line online help for further information.
## STACK LEVEL JUMPERS

The left three position of the J3 connector are used to select the stack level of the card:

<table>
<thead>
<tr>
<th>JUMPERS:</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>STACK LEVEL:</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>I2C ADDRESS:</td>
<td>0x18</td>
<td>0x19</td>
<td>0x1A</td>
<td>0x1B</td>
<td>0x1C</td>
<td>0x1D</td>
<td>0x1E</td>
<td>0x1F</td>
</tr>
</tbody>
</table>
POWER REQUIREMENTS

The MEGA-BAS card requires an external 24VDC/AC regulated power supply. Power is supplied to the board through the dedicated connector in the upper right corner (see BOARD LAYOUT). The boards accepts either DC or AC power source. If a DC power source is used, polarity is not important.

A local 5V regulator supplies up to 3A power to Raspberry Pi, and a 3.3V regulator powers the digital circuits. Isolated DC-DC converters are used to power the relays.

WE RECOMMEND USING ONLY THE 24VDC/AC POWER SUPPLY TO POWER THE RASPBERRY PI CARD

If multiple MEGA-BAS cards are stacked on top of each other, we recommend using a single 24VDC/AC power supply to power all the cards. The user must split the cable and run the wires to each MEGA-BAS card.

POWER CONSUMPTION:

- 50 mA @ +24V (all relays OFF)
- 150 mA @ +24V (all relays ON)
HARDWARE WATCHDOG

The MEGA-BAS card contains a built-in hardware watchdog which will guarantee that your mission-critical project will continue running even if Raspberry Pi software hangs up. After power up the watchdog is disabled, and becomes active after it receives the first reset.

The default timeout is 120 seconds. Once activated, if it does not receive a reset from Raspberry Pi within 2 minutes, the watchdog cuts the power and restores it after 10 seconds.

Raspberry Pi needs to issue a reset command before the timer on the watchdog expires. The command can be sent either on the I2C port, or by toggling GPIO13 (Pin 33 on the GPIO connector). The timer period after power up and the active timer period can be set from the command line. The number of resets is stored in flash and can be accessed or cleared from the command line. All the watchdog commands are described by the online help function.
ANALOG INPUTS/OUTPUTS CALIBRATION

All the analog inputs and outputs are calibrated at the factory, but firmware commands permit the user to re-calibrate the board, or to calibrate it to better precision. All inputs and outputs are calibrated in two points; select the two points as close to possible to the two ends of scale. To calibrate the inputs, the user must provide analog signals. (Example: to calibrate 0-10V inputs, the user must provide a 10V adjustable power supply). To calibrate the outputs, the user must issue a command to set the output to a desired value, measure the result and issue the calibration command to store the value.

The values are stored in flash and the input curve is assumed to be linear. If a mistake is made during calibration by typing the wrong command, a RESET command can be used to reset all the channels in the corresponding group to factory values. After RESET calibration can be restarted.

The board can be calibrated without a source of analog signals, by calibrating first the outputs and then routing the calibrated outputs to corresponding inputs. The following commands are available for calibration:

- **CALIBRATE 0-10V INPUTS:**   
  `megabas <id> cuin <channel> <value>`

- **RESET CALIBRATION OF 0-10V INPUTS:**   
  `megabas <id> rcuin`

- **CALIBRATE 10K INPUTS:**   
  `megabas <id> cresin <channel> <value>`

- **RESET 10K INPUTS:**   
  `megabas <id> rcresin`

- **CALIBRATE 0-10V OUTPUTS:**   
  `megabas <id> cuout <channel> <value>`

- **STORE CALIBRATED VALUE IN FLASH:**   
  `megabas <id> alta_comanda <channel> <actual_value>`

- **RESET CALIBRATION OF 0-10V OUTPUTS:**   
  `megabas <id> rcuout`
HARDWARE SPECIFICATIONS

ON BOARD RESETTABLE FUSE: 1A

0-10V INPUTS:

- Maximum Input Voltage: 12V
- Input Impedance: 20KΩ
- Resolution: 12 bits
- Sample rate: tbd

CONTAC CLOSURE INPUTS

- Maximum count frequency: 100 Hz

0-10V OUTPUTS:

- Minimum Output Load: 1KΩ
- Resolution: tbd

TRIAC OUTPUTS:

- Maximum Output Current: 1A
- Maximum Output Voltage: 120V
LINEARITY OVER FULL SCALE

Analog inputs are processed using 12 bit A/D converters internal to the on-board processor. The inputs are sampled at 675 Hz.

Analog outputs are PWM synthesized using 16 bit timers. PWM values range from 0 to 4,800.

All inputs and outputs are calibrated at test time at the end points and values are stored in flash.

After calibration we checked the linearity over full scale and obtained the following results:

<table>
<thead>
<tr>
<th>Channel</th>
<th>Max Error</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-10V IN</td>
<td>15μV</td>
<td>0.15%</td>
</tr>
<tr>
<td>0-10V OUT</td>
<td>10μV</td>
<td>0.1%</td>
</tr>
</tbody>
</table>
MECHANICAL SPECIFICATIONS

104 mm
21 mm 58 mm 25 mm

59 mm
10 mm
14 mm
7 mm 1.5 mm
SOFTWARE SETUP

1. Have your Raspberry Pi ready with the latest OS.

2. Enable I2C communication:

   ```
   ~$ sudo raspi-config
   ```

   1. Change User Password  Change password for default user
   2. Network Options     Configure network settings
   3. Boot Options        Configure options for start-up
   4. Localisation Options Set up language and regional settings to match..
   5. Interfacing Options  Configure connections to peripherals
   6. Overclock           Configure overclocking for your Pi
   7. Advanced Options    Configure advanced settings
   8. Update              Update this tool to the latest version
   9. About raspi-config  Information about this configuration

   P1 Camera       Enable/Disable connection to the Raspberry Pi Camera
   P2 SSH          Enable/Disable remote command line access to your Pi
   P3 VNC          Enable/Disable graphical remote access to your Pi using...
   P4 SPI          Enable/Disable automatic loading of SPI kernel module
   P5 I2C          Enable/Disable automatic loading of I2C kernel module
   P6 Serial       Enable/Disable shell and kernel messages to the serial port
   P7 1-Wire       Enable/Disable one-wire interface
   P8 Remote GPIO  Enable/Disable remote access to GPIO pins

3. Install the megabas software from github.com:

   ```
   ~$ git clone https://github.com/SequentMicrosystems/megabas-rpi.git
   ```

4. ```
   ~$ cd /home/pi/megabas-rpi
   ```

5. ```
   ~/megaioind-rpi$ sudo make install
   ```

6. ```
   ~/megaioind-rpi$ megabas
   ```

   The program will respond with a list of available commands.

   Type "megabas -h" for online help.

After installing the software, you can update it to the latest version with the commands:

   ```
   ~$ cd /home/pi/megabas-rpi
   ~$ megaioind-rpi$ git pull
   ~$ megaioind-rpi$ sudo make install
   ```
8 UNIVERSAL INPUTS  4 DIGITAL OUTPUTS  4 ANALOG OUTPUTS
* 0-10V  * TRIAC TBD  * 0-10V
* 1K/10K THERMISTOR  * DRY CONTACT

Input Select Jumpers

J72  J73  J42  J43  0-0  0-0  0-0  0-0
J74  J75  J44  J45  0-0  0-0  0-0  0-0

MOUNTING HOLES

H1  H2  H3  H4

SEQUENTMICROSYSTEMS.COM
USE PA0 TO READ BOARD IF FOR PROGRAMMING.
ID=0V FOR BAS