A-Vision Solutions – Raspberry Pi 12-24V Machine control card

General

This Raspberry Pi MOTOR interface has 3x MOSFET digital outputs (12-24V / max 2A), 2x TRIAC outputs (250V / 10A) and 7x MOSFET H-bridge MOTOR outputs (12-24V / max 2A).
Next to that, it has 32x 12-24V digital inputs with a pull-up resistor (level is HIGH when not connected) and 1x DS18B20 input (powered by 3.3V).

The MOSFET outputs are **not** isolated outputs. The 3x MOSFET digital outputs are protected with a 2A self-resetting fuse.

Furthermore, this interface board also supplies a 5V power to the Raspberry Pi.

All inputs and outputs have screw connectors

Preparations

For the 32x digital inputs the I^{2}C needs to be enabled (please refer to https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/configuring-i2c for instructions).

The DB18B20 is connected using the 1-wire protocol which also need to be enabled on the Raspberry Pi (please refer to https://learn.adafruit.com/adafruits-raspberry-pi-lesson-11-ds18b20-temperature-sensing/ds18b20 for instructions).

Used GPIO pins

The table below show what GPIO pins are used to control the outputs (BCM numbering).

Interface	Raspberry Pi		Comments
OUT1	GPIO13	Active HIGH	Floating output
OUT2	GPIO12	Active HIGH	Floating output
OUT3	GPIO6	Active HIGH	Floating output
220V1	GPIO27	Active HIGH	Active means TRIAC contact closed
220V2	GPIO17	Active HIGH	Active means TRIAC contact closed
M1 LEFT	GPIO11	Active HIGH	See table below
M1 RIGHT	GPIO8	Active HIGH	See table below
M2 LEFT	GPIO25	Active HIGH	See table below
M2 RIGHT	GPIO10	Active HIGH	See table below
M3 LEFT	GPIO16	Active HIGH	See table below
M3 RIGHT	GPIO26	Active HIGH	See table below
M4 LEFT	GPIO23	Active HIGH	See table below
M4 RIGHT	GPIO24	Active HIGH	See table below
M5 LEFT	GPIO19	Active HIGH	See table below
M5 RIGHT	GPIO18	Active HIGH	See table below
M6 LEFT	GPIO9	Active HIGH	See table below
M6 RIGHT	GPIO7	Active HIGH	See table below
M7 LEFT	GPIO22	Active HIGH	See table below
M7 RIGHT	GPIO5	Active HIGH	See table below
DS18B20	GPIO4	1-WIRE	See instructions below
PCF8575	GPIO2/GPIO3	I^{2}C	See instructions below

LEFT	RIGHT	OUTL	OUTR	Comments
L	L	X	X	Both outputs floating / motor free run
L	H	VCC	GND	Motor turning RIGHT
H	L	GND	VCC	Motor turning LEFT
H	H	GND	GND	Motor stalled

Table 1 Motor bi-directional control

Installation

The Raspberry Pi 12-24V MOTOR interface HAT has a 40p header with mounting holes that can easily be placed on top of a Raspberry Pi.

Warnings

Please make sure the 12-24V power supply polarity is connected correctly, incorrect polarity may damage the card.

Please do not touch the control card while powered on. MOSFETs are highly sensitive and can easily getting damaged if touched. Also, the 220V connected on the board can cause an electric shock.

Driver software

A Python library for reading and writing the 32x additional I/O via the (2x) PCF8575 is available at https://github.com/A-Vision-Software/PCF8575.

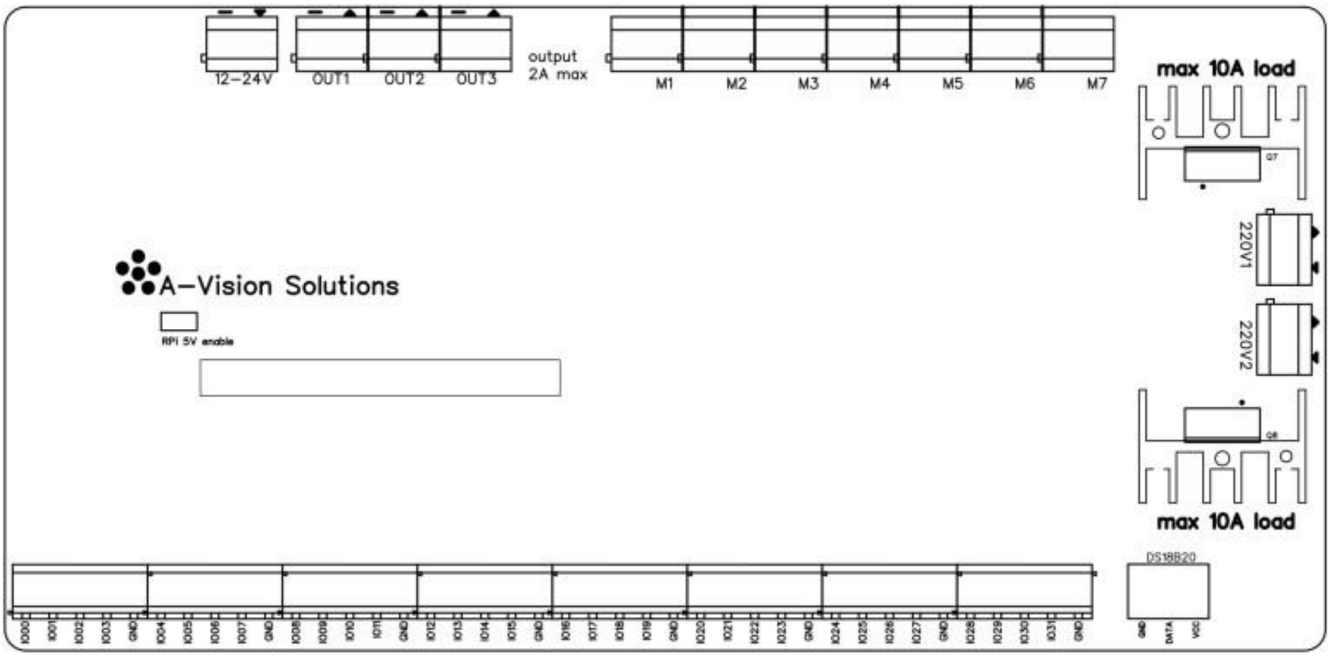
To read the DB18B20 temperature sensor the w1thermsensor Python library can be used. Please follow the instructions from https://pypi.org/project/w1thermsensor/ to install.

PCB layout

The PCB layout is shown below.

Make sure to connect the power supply with the right polarity.

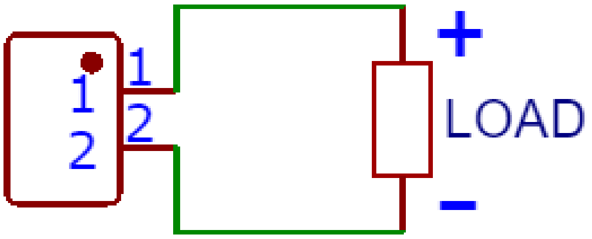
The 2A digital outputs are so-called floating outputs. This means that when **inactive**, it is an open end.



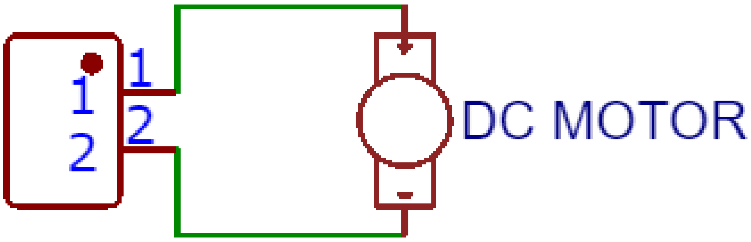
Connecting diagrams

Please follow the instructions below for connecting hardware to the control card.

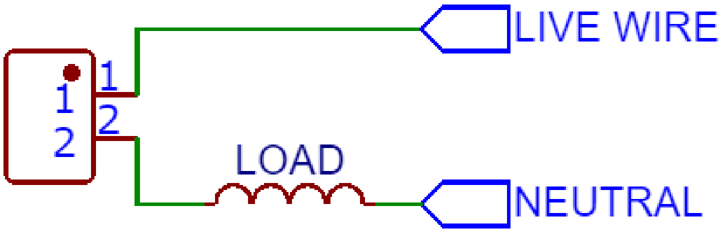
2A Digital outputs



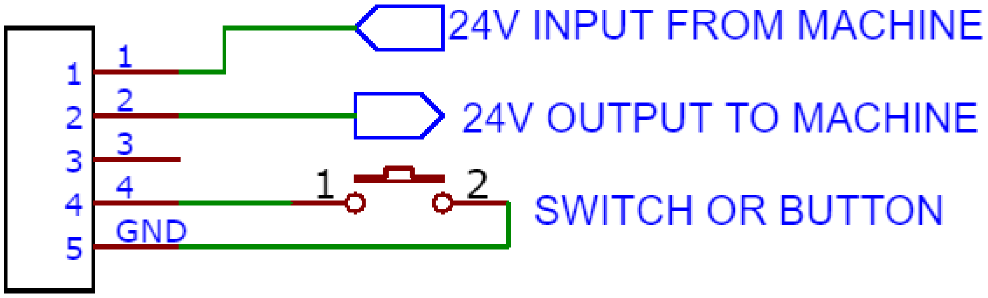
2A H-Bridge motor outputs



220V/10A TRIAC switch



DIGITAL I/O



Please note that a switch/button needs to be connected between the input and ground (GND).
This means that for switches/buttons the active level is LOW.
For each 4 inputs there is also a GND connection available.

Connecting a DS18B20 temperature sensor

The DS18B20 sensor connector has 3 pins, GND, DATA and VCC that corresponds to the DS18B20 sensor wires.

Software

Example software is provided using the drivers mentioned before.

Please note the following for programming the 32x digital I/O

```
        # initializing PCF I/O expander devices
        PCF1 = PCF8575(1, 0x20)       # where 1 is the used I2C port and 0x20 is the first PCF device address
        PCF2 = PCF8575(1, 0x21)       # where 1 is the used I2C port and 0x21 is the second PCF device address

        # PCF1 is connected to IO00 to IO15 on the card
        # PCF2 is connected to IO16 to IO31 on the card

        # Reading inputs
        INP1 = PCF1.port[0]           # PCF1.port[0] refers to IO00 on the card
        INP2 = PCF1.port[8]           # PCF1.port[8] refers to IO08 on the card
        INP3 = PCF2.port[0]           # PCF2.port[0] refers to IO16 on the card
        INP4 = PCF2.port[8]           # PCF2.port[8] refers to IO24 on the card

        # Writing outputs
        PCF1.port[3] = True           # PCF1.port[3] refers to IO03 on the card
        PCF1.port[11] = True          # PCF1.port[11] refers to IO11 on the card
        PCF2.port[3] = True           # PCF2.port[3] refers to IO19 on the card
        PCF2.port[11] = True          # PCF2.port[11] refers to IO27 on the card
```