Motor box

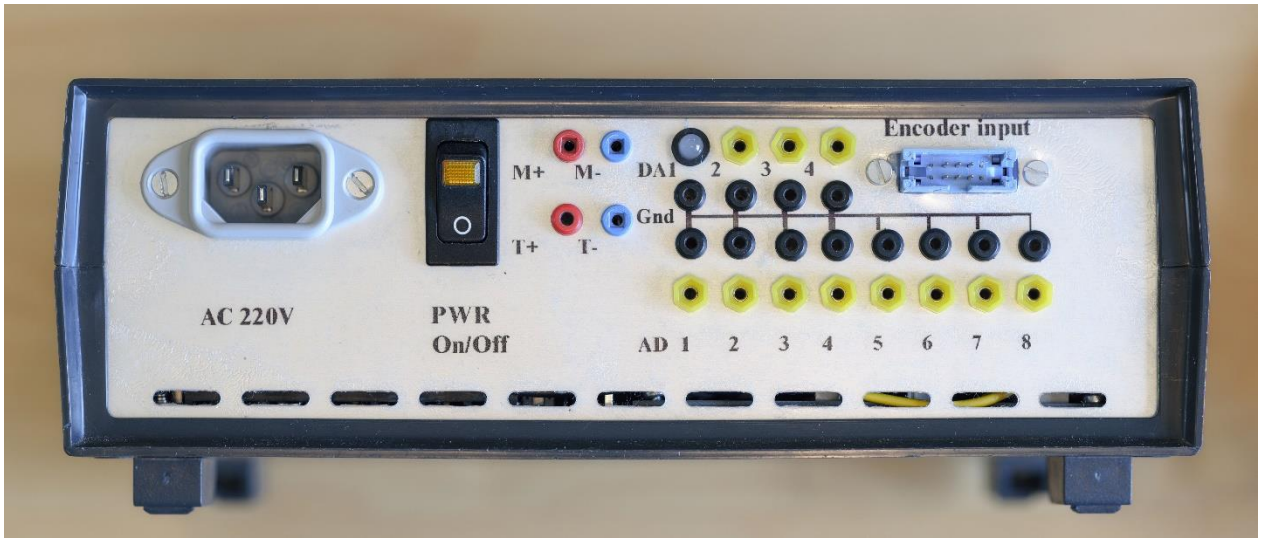
User guide

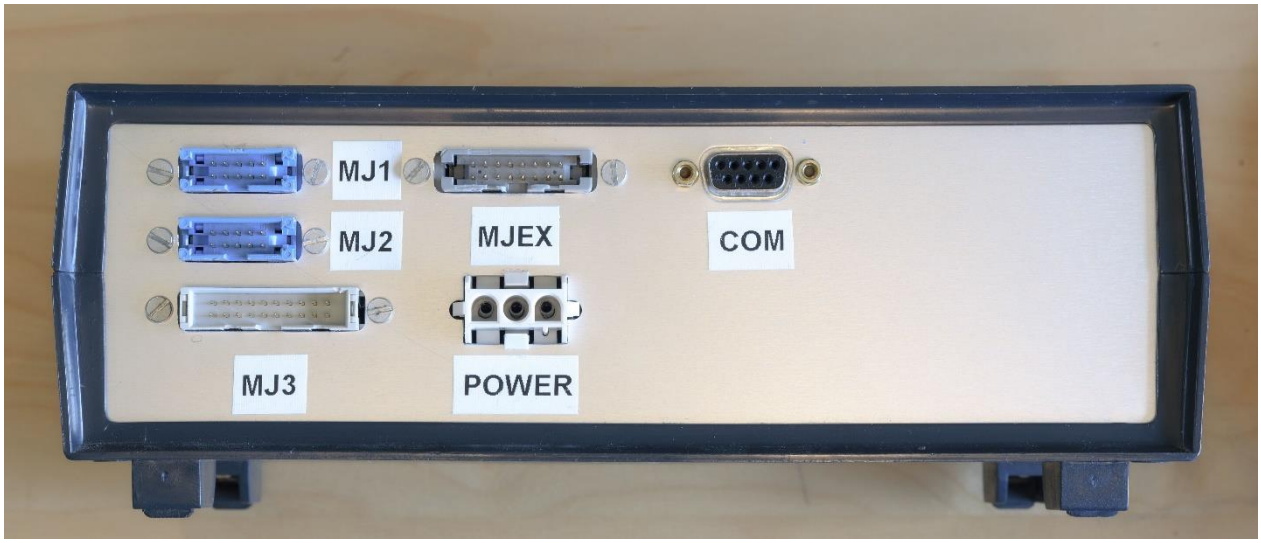
For use with the Elevator model and
Ping-pong board at the real-time lab

1 Contents

2 Introduction

This document describes the features and operation of the motor box used to power and control the DC motors connected to the elevator model and ping-pong-boards at the real-time lab.





3 General

3.1 Functionality

The motor box is used to power a 12V motor, and read its connected quadrature encoder. The box reads signals for motor enable, direction, and speed, as well as control signals for the encoder value output.

3.2 Modes of operation

The motor box has two modes of operation:

- **Normal** (orange status LED):
  For use with the ping-pong board.
  The motor will run at any speed the user requests on the analog speed input pin.
- **Brake Threshold** (blue status LED):
  For use with the elevator.
  The motor will brake and stay stopped at speeds below a certain voltage threshold.

The motor brake can be manually engaged by setting the motor enable pin low, regardless of mode.

See section 4.1 (Headers) for details on how to control the motor.

3.3 Motor

The motor is attached to the M+ and M- pins. Swapping these only changes the direction of the motor. Motor control inputs (enable, direction, analog speed) are converted are converted by the internal microcontroller and motor driver into a differential analog signal up to 12V DC.

3.4 Encoder

As quadrature encoders count incrementally, there is a possibility that the accumulated value will overflow when counting over long distances. The motor box both has an encoder reset pin to set the encoder count to 0, and a configurable encoder divisor that lowers the resolution of the encoder.

It should typically not be necessary to change the encoder divisor. Recommended usage is to move the motor to a known position and reset the encoder during some startup procedure.

3.4.1 Encoder divisor & timing

The encoder is read by triggering pin change interrupts from changes to either both the encoder disc's channels, or just one of the channels. Setting the divisor to 0 will make the motor box trigger interrupts on both channels, giving double resolution. A divisor of 1 or above will use a single channel.

Because of the use of interrupts, there is a maximum frequency where the motor box will spend too much CPU time on the encoder for it to be responsive to other tasks. It is not recommended to go above these frequencies

- `divisor == 0  :  115k double-channel changes / second`
- `divisor >= 1  :   95k single-channel changes / second`
  `= 190k double-channel changes / second`

You can find the number of double-channel changes per second by setting the encoder divisor to 0, reading the encoder to find the number of counts between the two maximum positions, and dividing by the fastest time taken between these positions.

4 User interface

4.1 Headers

All pins operate at a 5V level.

Header MJ1 and MJEX contain only input pins (and one ground pin), and MJ2 only output pins (and one ground pin). The unused pins are physically disconnected inside the motor box.

To move the motor, set the motor enable (MJ1 EN) pin high, choose the direction with the direction pin (MJ1 DIR), and set the speed with the analog speed input pin (MJEX A0).

Setting the motor enable (MJ1 EN) pin low will engage the motor brake, causing the motor to come to a rapid stop. If the motor box is in Brake Threshold mode, setting the speed below the configured threshold will also engage the motor brake.

All pins on MJ1 have internal pull-ups enabled. If MJ1 and MJEX are disconnected, the motor will typically slowly start to drift in the positive direction (up/right).

The encoder value is a signed 16-bit integer (`int16_t`), and is read out from MJ2. The byte select pin (MJ1 SEL) selects whether the high or low byte is written to MJ2. Setting the output enable pin (MJ1 !OE) low will sample and hold the encoder value, such that it does not change while the encoder value is being read by the attached device.

The normal procedure for reading the encoder is as follows:

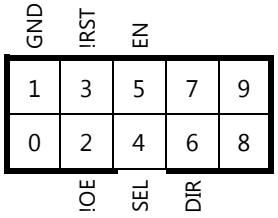
1: Set !OE low, to sample and hold the encoder value
2: Set SEL low to output high byte
3: Wait approx. 20 microseconds for output to settle
4: Read MJ2 to get high byte
5: Set SEL high to output low byte
6: Wait approx. 20 microseconds
7: Read MJ2 to get low byte
8: Set !OE to high

The encoder reset pin (MJ1 !RST) will reset the internal counter to 0 when pulled low.

Note that if you want to reset the encoder after each read operation (to read deltas, instead of absolute value), the encoder value will likely have changed in the time between the value has been read and when it was reset. In order to minimize this error, you should insert the encoder reset operation as early as after step 3 (the first delay), and possibly also tweak this delay to be as short as possible (expected fastest stable delay is about 16 us).

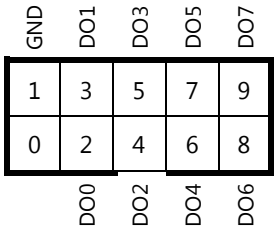
It is typically better to calculate this delta on the attached device, and instead configure an appropriate encoder divisor such that the encoder value never overflows.

4.1.1 MJ1 (CTRL_SIGNAL_IN)



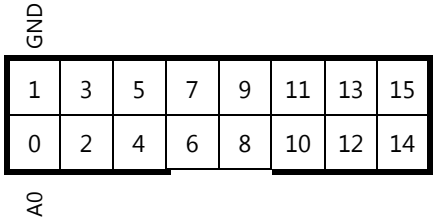
Pin	Name	Description
1	GND	Common ground
2	!OE	Encoder output enable (inverted)
3	!RST	Encoder reset (inverted)
4	SEL	Encoder select low byte
5	EN	Motor enable (motor brake disable)
6	DIR	Motor direction up (elevator) / right (pong)

4.1.2 MJ2 (COUNTER_OUT)



Pin	Name	Description
1	GND	Common ground
2 .. 9	DOn	Encoder byte output

4.1.3 MJEX (ANALOG_IN)



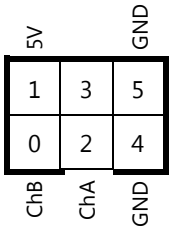
Pin	Name	Description
0	A0	Motor speed analog input (0-5V)
1	GND	Common ground

4.2 Motor

4.2.1 Power (M+, M-)

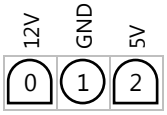
M+ and M- should be connected to the motor. The voltage is differential, so swapping the pins only changes the direction of the motor.
T+ and T- are not connected internally.

4.2.2 Encoder input



4.3 Elevator panel power

The elevator button panel gets its power from the 3-pin power connector on the back.



4.4 COM port

The COM port allows for basic diagnostics and configuration for the motor box. Terminal configuration: 115200 baud, 8 bit, 1 stop bit, no parity.

All commands are a single character, optionally followed by a single number (between 0 and 255), then a newline (`\n`). Settings are written to EEPROM, and will persist when the device is powered off.

Cmd	# params	Description	Range
`t`	0 or 1	Read/write motor brake threshold. Set `t0` for Normal mode	0..1
`q`	0 or 1	Read/write motor quick responsiveness mode	0..1
`s`	0 or 1	Read/write motor speed multiplier	1..100
`d`	0 or 1	Read/write encoder divisor	0..255
`a`	0	Read analog input	
`e`	0	Read encoder value	
`r`	0	Reset encoder to 0	

4.5 Alarms & Status (RGB LED & Buzzer)

The pins on MJ1, MJEX, and the encoder input are somewhat protected and can tolerate up to +20V and to some extent negative voltages down to -5V. An alarm will alert when the user has connected a too high voltage to a pin (above ~6V).

The output pins (MJ2) are also protected against short-circuits and applied voltages. An alarm will alert when these pins are externally driven (eg. if anybody tries to force a logical 1 when the motor box outputs a logical 0, or vice versa).

When the motor box is powered on, it will self-test the LED and buzzer by briefly showing a white status and playing one beep of each of the alarm pitches. It is not recommended to use the motor box if these error-reporting mechanisms are not functional.

After this test, the LED will show either steady orange or blue, depending on the mode of operation.

Status	**RGB LED color**	**Buzzer**
Normal operation (smooth mode)	Orange	No sound
Normal operation (quick mode)	Yellow	No sound
Threshold brake (smooth mode)	Light blue	No sound
Threshold brake (quick mode)	Blue	No sound
Pin input voltage out of range	Magenta	High pitch
MJ2 externally driven	White	Low pitch

4.6 Power (AC 220V)

The box requires an AC 220V 50Hz power connection. The motor box will start operation after the LED and buzzer power-on self-test.

Turning the motor box on for less than 5 seconds two times in a row will trigger a mode switch on the third bootup, which will also set these default values to all the configuration parameters:

Parameter	**Normal mode** (autumn)	**Threshold brake** (spring)
Motor brake threshold	0	5
Motor quick responsiveness	0 (smooth)	1 (quick)
Motor speed multiplier	85	100
Encoder divisor	0	9

5 Changelog

5.1 Version 1 (2010)

Made by:
Ole Johnny Borgersen
Marius Lind Volstad

5.2 Version 2 (2021)

Made by:
Anders Rønning Petersen

5.2.1 Firmware changes

- Removed unused features:
  - Configuration & COM port options
    - Alarm enable/disable
    - Manual motor move
    - PID controller
    - Buzzer volume
- Fixed broken functionality:
  - Incorrect encoder output, resulting from CPU over-utilization due to several interrupt sources triggering too frequently
  - Motor brake enabling at <0.005V MJEX-A0 input, resulting in elevators sometimes not stopping properly, and pong controllers behaving jittery
- Added new features:
  - Brake Threshold mode, and fixed Normal mode
  - Encoder divisor
  - Motor responsiveness (current decay mode)