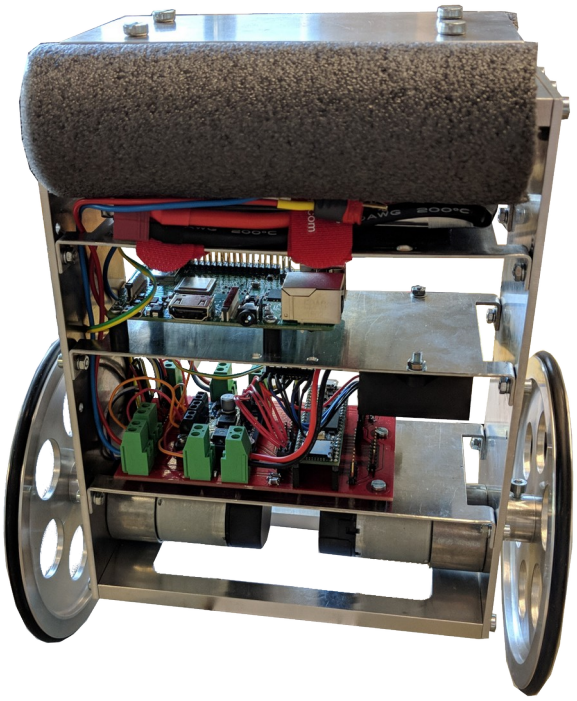
User Manual: Building and using a Wheeled Inverted Pendulum

This document is concerned with building and using a small balancing robot, with resources available under: *https://github.com/denglerchr/Wheeled-Inverted-Pendulum*



Authors:
Christian Dengler

Table of Contents

I. Components and electronics setup

I.1. Components

The components are relatively inexpensive and the setup is modular, as such, even if a part is not available anymore, it should be possible to upgrade the Raspberry. The wheeled inverted pendulum (WIP), as of early 2020, consists of the following parts:

Energy supply:

- 12V (11.1V) battery

Actuators:

- DC Motors with Encoder

Sensors:

- Accelerometer+gyroscope, type MPU6050
- (Encoders on the motors)

Computing Units:

- Arduino IDE compatible microcontroller board: Teensy 3.6
- Single board compute, Raspberry PI 3B+

Further Parts:

- Dual H-bridge from HobbyTronics
- CircuitBoard, custom made, files given in the Github repo
- DCDC converter to get 5V and 3.3V required for some parts

I.2. General Setup

The WIP contains two computing units. The Teensy is a microcontroller that is real time capable. The Raspberry Pi is more of a computer type and brings increased computing pwoer, but is unsuited for precise timings. As such, the Teensy is the master, giving the timing, whereas the Raspberry Pi is just a computing unit, reacting to the calls from the Teeny. The communication between both is serial (UART).

The main communication loop on the Teensy is as follows

1. The Teensy board reads out the encoder values and the gyroscope and the accelerometer values.
2. The Teensy transforms the sensor data into a state vector, containing the position, velocity as well as angles and angular velocities.

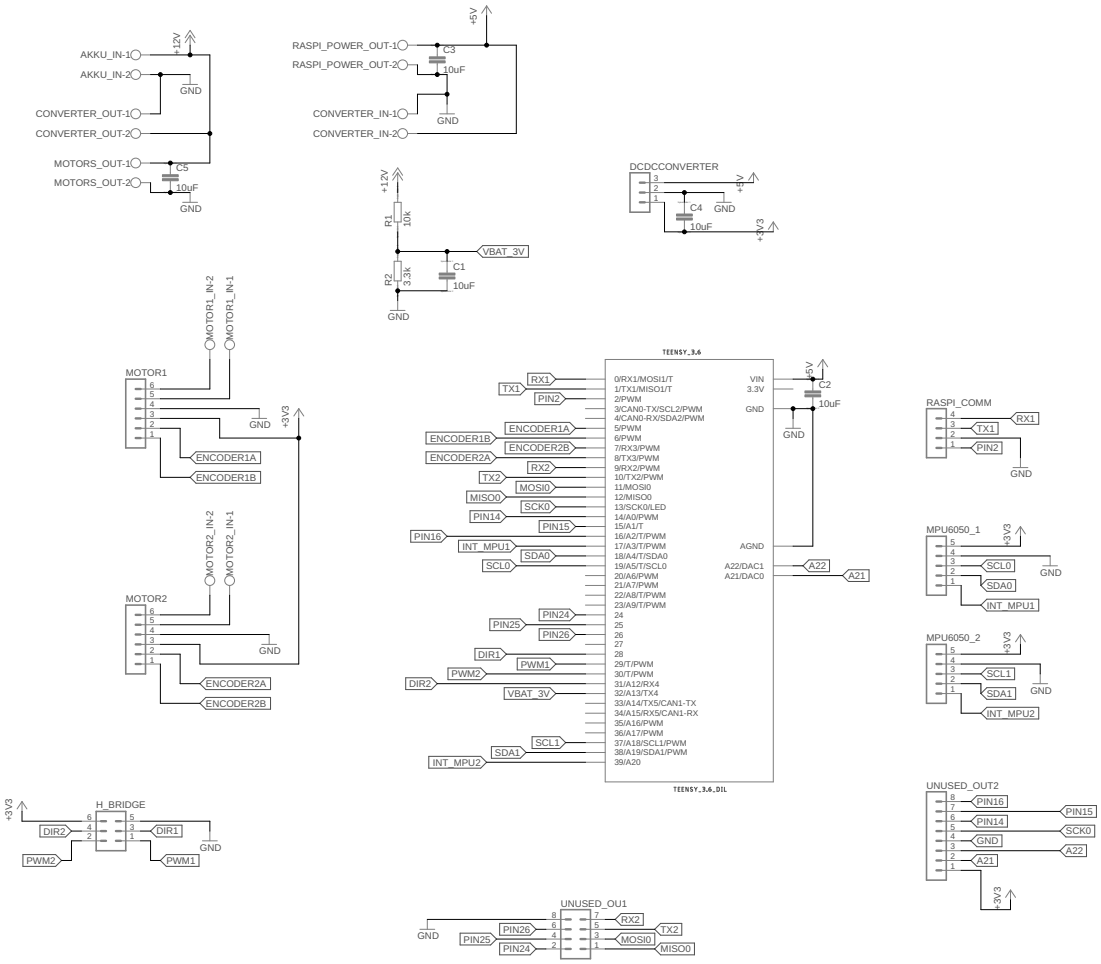
3. The Teensy reads the input buffer (i.e. the data the Raspberry Pi sent last cycle) and sends the state vector to the Raspberry Pi. If no data is available from the Raspberry Pi, the teensy will not send and wait, it will however always keep updating its state vector each cycle.
4. The Teensy can send some information to a PC at this point, if the specific line is uncommented, this is usefull for debugging and testing.

The main communication loop on the Raspberry Pi is:

1. Wait for data, i.e., to have 7 Float32 values = 7*4 bytes of data.
2. Compute control law and save state and action in a Matrix.
3. Send 2*4 bytes to the Teensy board and go into a waiting state until the next state vector is received. If enough time passed, print to PC.

If Keyboard-Interrupt → stop main cycle and save the data matrix in a hdf5 file

I.3. Schematics



05.07.18 14:44 f=0.72 /home/christiand/Dropbox/Uni/Projekte/2018_SegwayUpgrade/01_Schaltplan/Platine/V2/main.sch (Sheet:

II. Setting up the Teensy

III. Setting up the Raspberry PI

III.1. Preparing your computer and the microcontroller (Linux Version)

III.2. Preparing the Raspberry Pi

IV. Starting a control algorithm