

1.1 Vedlegg 1 Universal Robots Real Time Date Exchange Eksempel

```
#!/usr/bin/env python
# Copyright (c) 2016, Universal Robots A/S,
# All rights reserved.
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are
# met:
#
# * Redistributions of source code must retain the above copyright
#   notice, this list of conditions and the following disclaimer.
# * Redistributions in binary form must reproduce the above copyright
#   notice, this list of conditions and the following disclaimer in the
#   documentation and/or other materials provided with the distribution.
# * Neither the name of the Universal Robots A/S nor the names of its
#   contributors may be used to endorse or promote products derived
#   from this software without specific prior written permission.
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
# IS" AND
# ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
# IMPLIED
# WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED. IN NO EVENT SHALL UNIVERSAL ROBOTS A/S BE LIABLE FOR ANY
# DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
# DAMAGES
# (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
# SERVICES;
# LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
# AND
# ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
# TORT
# (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
# THIS
# SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

import sys
sys.path.append('.')
import logging

import rtde.rtde as rtde
import rtde.rtde_config as rtde_config

#logging.basicConfig(level=logging.INFO)

ROBOT_HOST = 'localhost'
ROBOT_PORT = 30004
config_filename = 'control_loop_configuration.xml'

keep_running = True

logging.getLogger().setLevel(logging.INFO)

conf = rtde_config.ConfigFile(config_filename)
state_names, state_types = conf.get_recipe('state')
setp_names, setp_types = conf.get_recipe('setp')
watchdog_names, watchdog_types = conf.get_recipe('watchdog')

con = rtde.RTDE(ROBOT_HOST, ROBOT_PORT)
con.connect()

# get controller version
con.get_controller_version()
```

```

# setup recipes
con.send_output_setup(state_names, state_types)
setp = con.send_input_setup(setp_names, setp_types)
watchdog = con.send_input_setup(watchdog_names, watchdog_types)

# Setpoints to move the robot to
setp1 = [-0.12, -0.43, 0.14, 0, 3.11, 0.04]
setp2 = [-0.12, -0.51, 0.21, 0, 3.11, 0.04]

setp.input_double_register_0 = 0
setp.input_double_register_1 = 0
setp.input_double_register_2 = 0
setp.input_double_register_3 = 0
setp.input_double_register_4 = 0
setp.input_double_register_5 = 0

# The function "rtde_set_watchdog" in the "rtde_control_loop.urp" creates a
1 Hz watchdog
watchdog.input_int_register_0 = 0

def setp_to_list(setp):
    list = []
    for i in range(0,6):
        list.append(setp.__dict__["input_double_register_%i" % i])
    return list

def list_to_setp(setp, list):
    for i in range (0,6):
        setp.__dict__["input_double_register_%i" % i] = list[i]
    return setp

#start data synchronization
if not con.send_start():
    sys.exit()

# control loop
while keep_running:
    # receive the current state
    state = con.receive()

    if state is None:
        break;

    # do something...
    if state.output_int_register_0 != 0:
        new_setp = setp1 if setp_to_list(setp) == setp2 else setp2
        list_to_setp(setp, new_setp)
        # send new setpoint
        con.send(setp)

    # kick watchdog
    con.send(watchdog)

con.send_pause()

con.disconnect()

```

```
<?xml version="1.0"?>
<rtde_config>
  <recipe key="state">
    <field name="target_q" type="VECTOR6D"/>
    <field name="target_qd" type="VECTOR6D"/>
    <field name="output_int_register_0" type="INT32"/>
  </recipe>

  <recipe key="setp">
    <field name="input_double_register_0" type="DOUBLE"/>
    <field name="input_double_register_1" type="DOUBLE"/>
    <field name="input_double_register_2" type="DOUBLE"/>
    <field name="input_double_register_3" type="DOUBLE"/>
    <field name="input_double_register_4" type="DOUBLE"/>
    <field name="input_double_register_5" type="DOUBLE"/>
  </recipe>

  <recipe key="watchdog">
    <field name="input_int_register_0" type="INT32"/>
  </recipe>
</rtde_config>
```