



This lecture will be recorded



C O M P A S

064-0026-00L: COMPAS II

Introduction to Computational Methods for Digital
Fabrication in Architecture

```
mesh.mesh.vertices[...]  
if callback:  
    if not callable(callback):  
        raise Exception('Callback is not callable.')  
    mesh.vertices[...]  
    mesh.vertices[...]  
    for k in range(kmax):  
        mesh.vertices[...]  
        for key in mesh.vertices():  
            if key in fixed:  
                continue  
            p = key_xyz[key]  
            nbs = mesh.vertex_neighbours(key, ordered=True)  
            c = center_of_mass_polygon([key_xyz[nbr] for nbr in nbs])  
            # update  
            attr = mesh.vertex[key]  
            attr['x'] += d * (c[0] - p[0])  
            attr['y'] += d * (c[1] - p[1])  
            attr['z'] += d * (c[2] - p[2])  
        if callback:  
            callback(mesh, k, callback_args)  
  
def smooth_mesh_length(mesh, lmin, lmax, fixed=None, kmax=100):  
    if callback:  
        if not callable(callback):  
            raise Exception('Callback is not callable.')  
        fixed = fixed or []  
        fixed = set(fixed)  
        for k in range(kmax):
```

slides and code

*[https://](https://tiny.cc/compas-ii)**tiny.cc/compas-ii***

TODAY

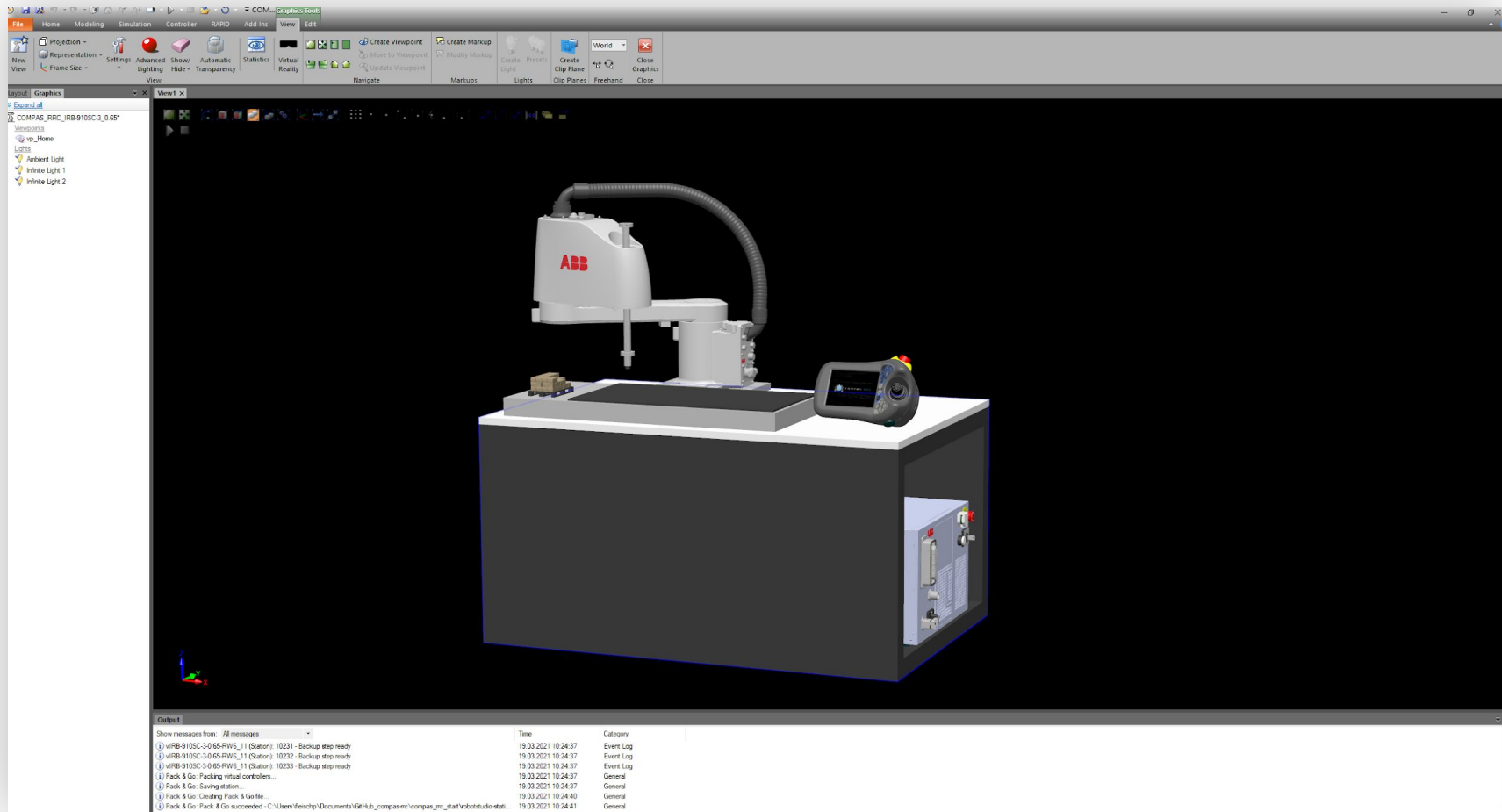
robot control


compas rrc

instructions overview

Today's goal

Understand **robot control modes** and **RRC basic set of instructions**




 Search or jump to... / Pull requests Issues Marketplace Explore


compas-rrc / compas_rrc_start Private Watch 2 Star 0 Fork 0


<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings


main 1 branch 0 tags


Go to file Add file + Code -


 fleischp pack and go updated with black background b12dcba 15 minutes ago 40 commits


 .vscode pack and go updated with black background 15 minutes ago


 Python first brick example created (new pack&go, new viewer and new python c... 3 days ago

 docker first structure done 11 days ago

 robotstudio-extras names changed agording to the course 11 days ago

 robotstudio-stations pack and go updated with black background 15 minutes ago

 robotstudio-viewers first brick example created (new pack&go, new viewer and new python c... 3 days ago

 README.md Update README.md 6 days ago

README.md

Welcome to COMPAS RRC

Online control for ABB robots over a simple-to-use Python interface.

Requirements

- Windows 10 Pro 64-bit
- CPU 2 GHz
- Memory 8 GB
- Disk 10 GB SSD
- Screen 1920x1080

Installation

Anaconda

- Anaconda 3:** Anaconda is an open source scientific Python distribution. With this tool, we can easily create a Python environment. Install Anaconda using default options.

Docker

- Docker Desktop:** Docker is a virtualization platform. We use it to run Linux containers for ROS on Windows machines.
- After installation, it is required to enable "Virtualization" on the BIOS of the computer. Usually this requires rebooting your computer and pressing a vendor-specific key (F2 ... F4 ... del ... are typical options) to enter the BIOS.

About

No description, website, or topics provided.

Readme


Releases


No releases published
Create a new release

Packages

No packages published
Publish your first package

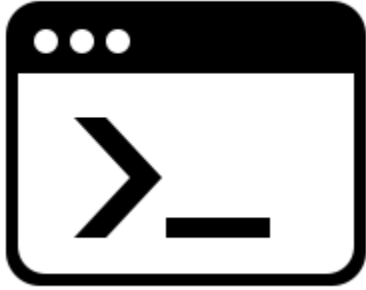
Contributors 2

 fleischp Philippe Fleischmann

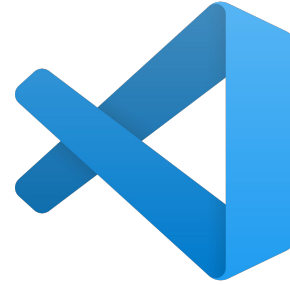
 gonzalocasas Gonzalo Casas

Languages

Python 100.0%



`docker-compose up -d`



Right-click → Compose Up

docker/rrc_virtual_controller

COMPAS RRC driver for ROS

robot control

compas rrc

instructions overview

Traditional programming (offline control)

ROBOT CONTROLLER



Upload robot code

LAPTOP

Vendor-specific
language and tools

ABB: RAPID

Stäubli: VAL3

UR: URScript

Kuka: KRL

Our goal: online control

ROBOT CONTROLLER



driver

LAPTOP



Overview: all the moving parts

ROBOT CONTROLLER



ROS

LAPTOP

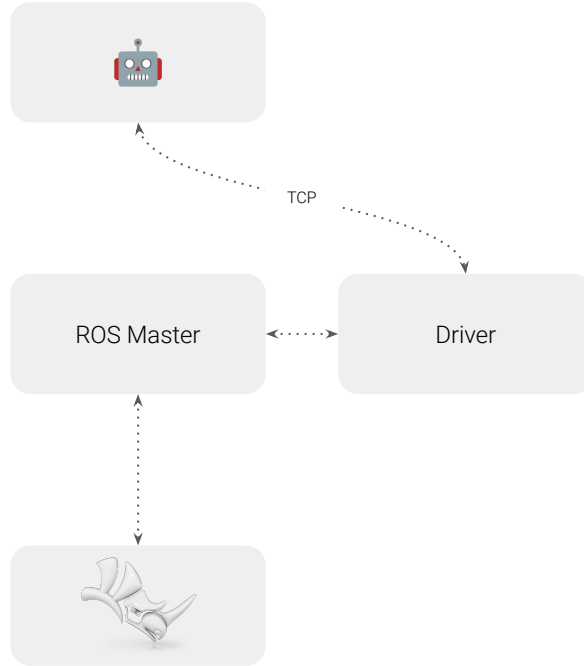


Overview: all the moving parts

ROBOT CONTROLLER



LAPTOP

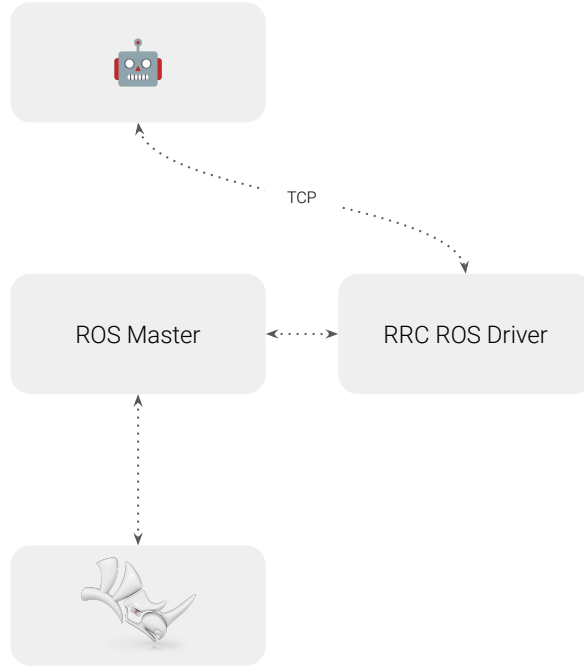


Overview: all the moving parts

ROBOT CONTROLLER



LAPTOP

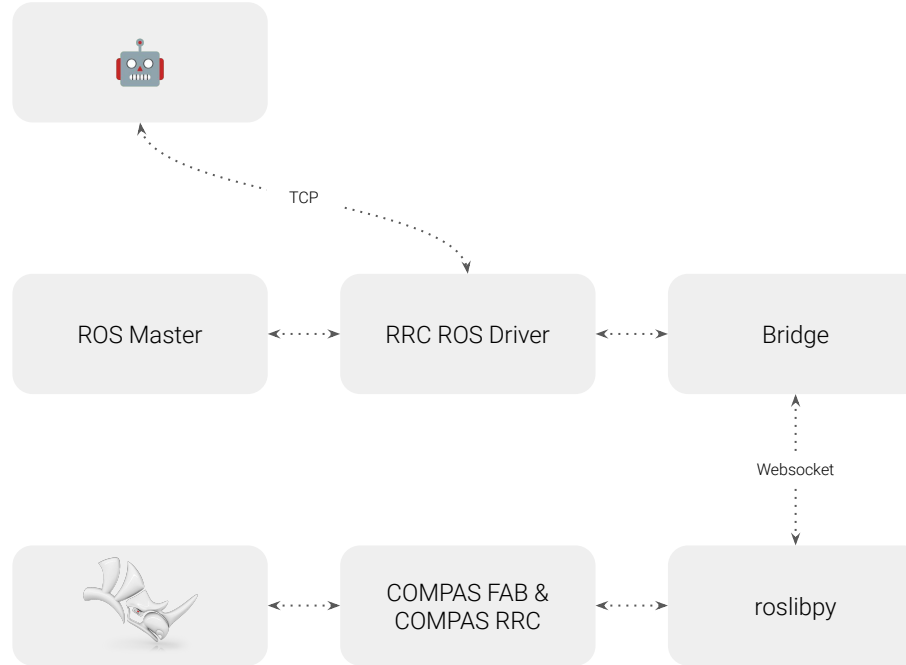


Overview: all the moving parts

ROBOT CONTROLLER



LAPTOP

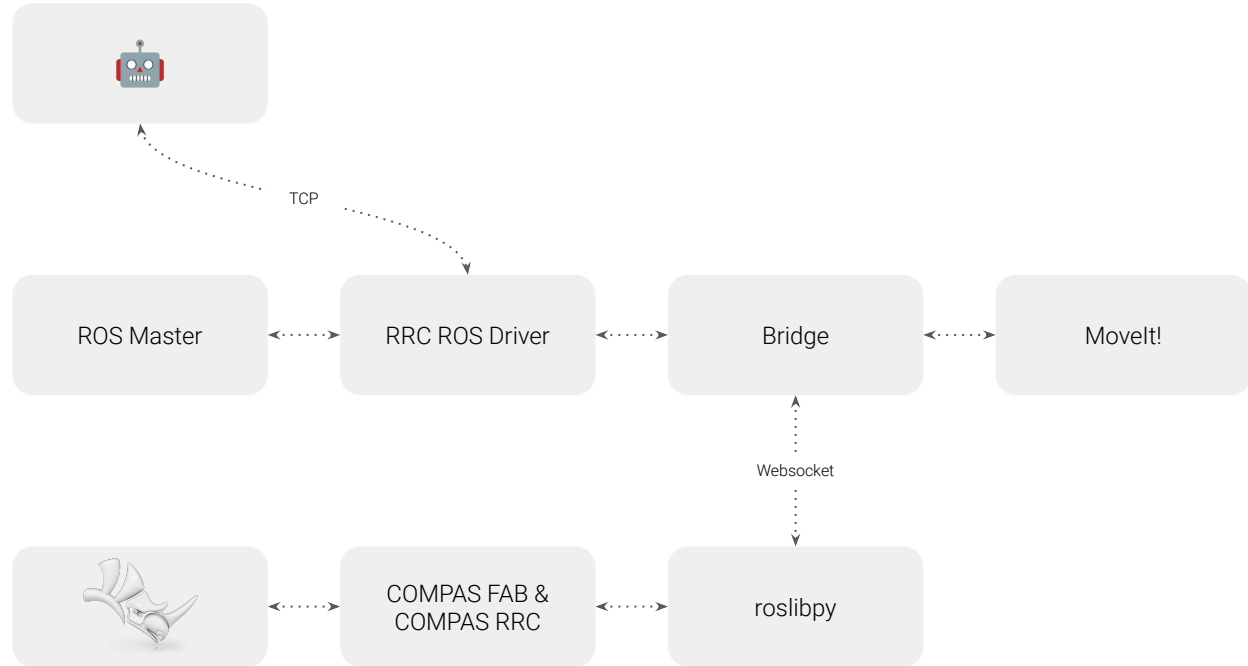


Overview: all the moving parts

ROBOT CONTROLLER



LAPTOP



Control



C O M P A S R R C

Offline control

Online real-time control

Online non-real-time control

robot control
compas rrc
instructions overview



C O M P A S R R C

COMPAS RRC





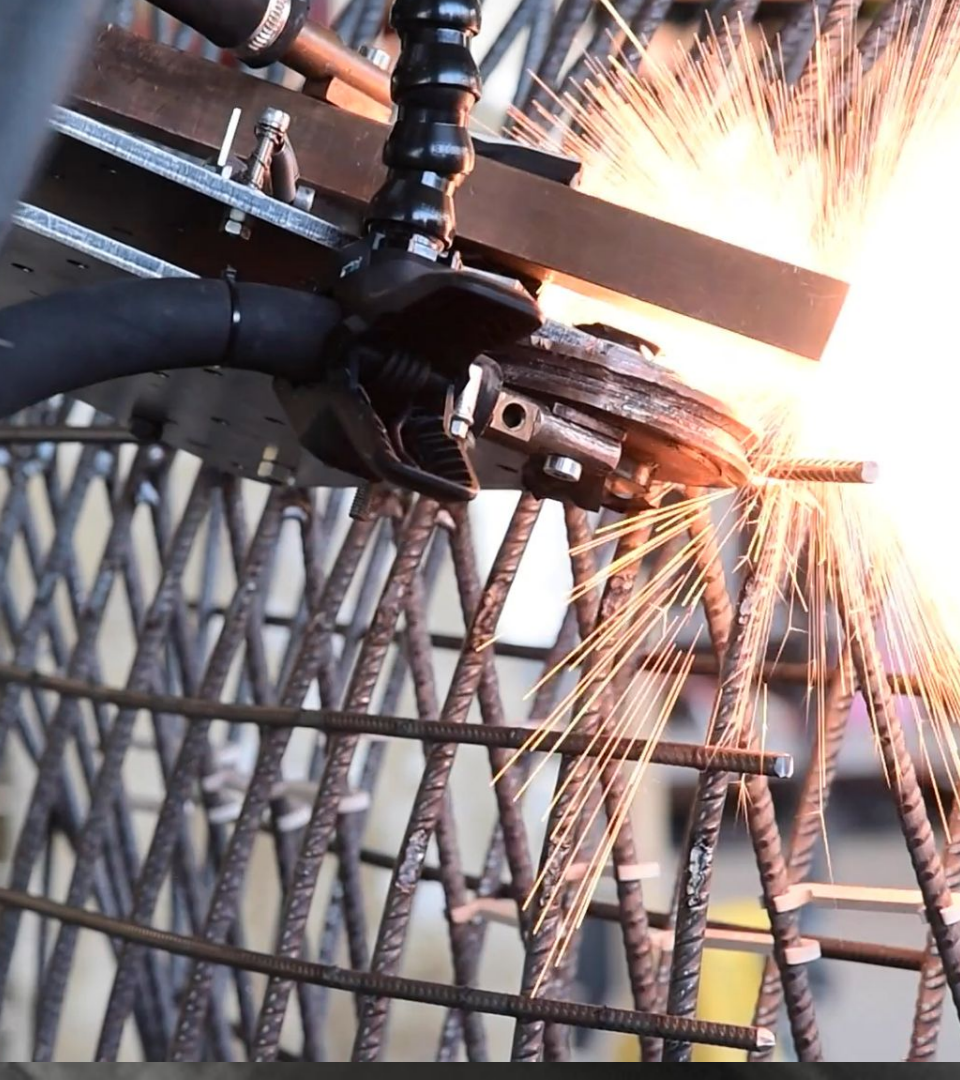
3D Concrete Printing

- Data streaming
- More than 300'000 points
- More than 4 hours of printing



WAAM

- Process Feedback
- Integrated Fronius welder
- Arc Weld PowerPac
- SmartTac



Mesh Mould

- Advanced Processing
- Slice positioning with
 - COMPAS RRC
- Rebar welding with
 - Externally Guided Motion

COMPAS RRC

Features

Live communication

Multi-Tasking

Multi-Move

Multi-Controller

Multi-Location

COMPAS RRC

Communication

Send ~75ms

Send and Wait ~150ms

Send and Wait in the Future

Send and Subscribe ~75ms

COMPAS RRC

Instructions

Motions

Signals

Basics

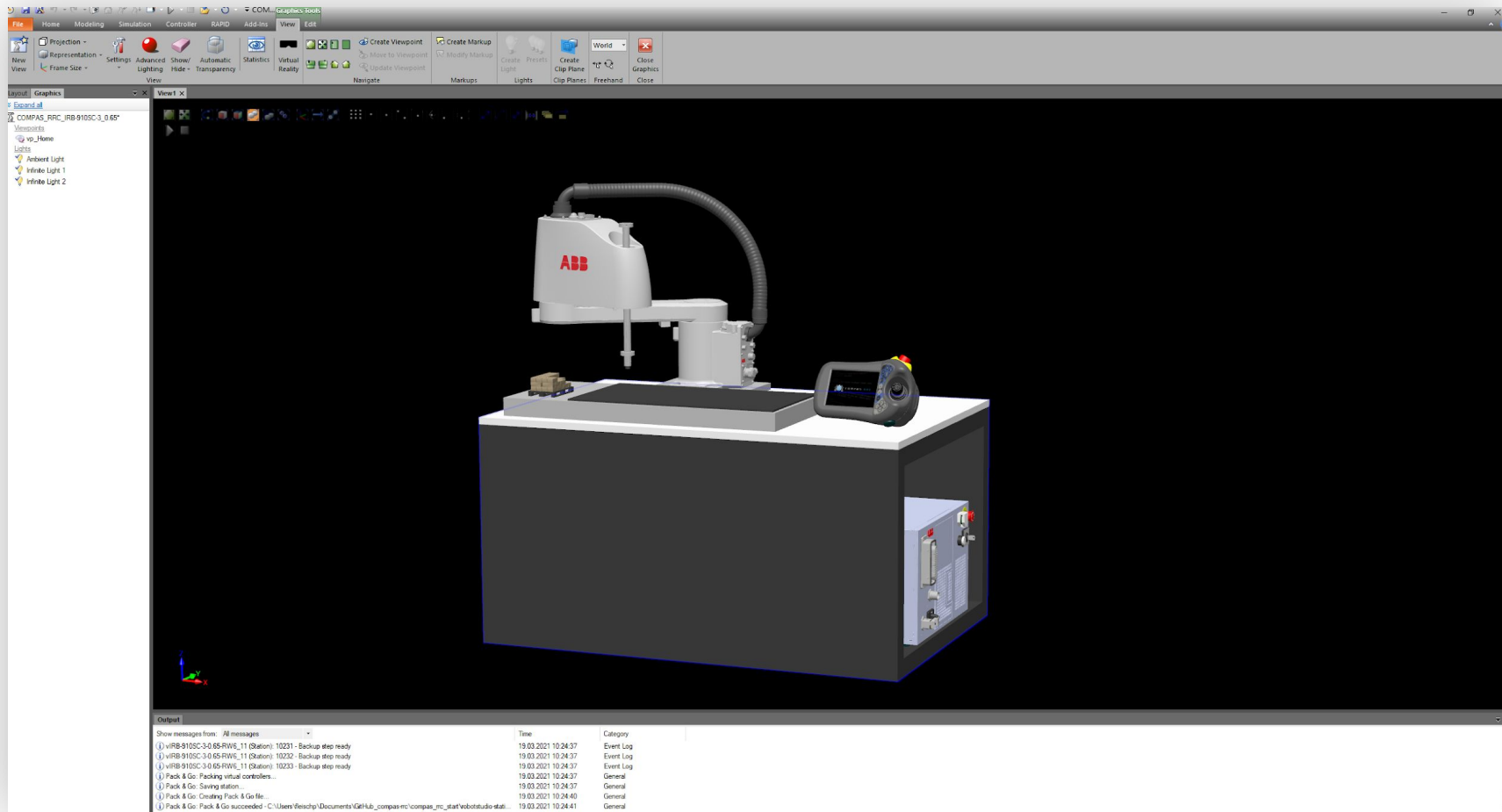
Utilities

Custom

robot control

compas rrc

instructions overview





Hello world

```
ros = rrc.RosClient()  
abb = rrc.AbbClient(ros, '/rob1')  
done = abb.send_and_wait(rrc.PrintText('Welcome to COMPAS_RRC'))
```



Send instruction (non-blocking)

```
# Send an instruction without waiting for any kind of feedback  
abb.send(rrc.PrintText('Hello.'))
```



Send instruction with feedback (blocking)

```
# Send and wait  
done = abb.send_and_wait(rrc.PrintText('Sent with feedback.'))
```




Send instruction with feedback (non-blocking)

```
# Send and defer waiting
future = abb.send(rrc.PrintText('feedback', feedback_level=rrc.FeedbackLevel.DONE))

# Here you can do other stuff [...]

# Wait for feedback
done = future.result(timeout=3.0)
```

COMPAS RRC

Basics

Set Tool

Set WorkObject

Set Max Speed

Set Acceleration

COMPAS RRC

Motion

Get Frame

Move to Frame

Get Joints

Move to Joints

Get Robtarget

Move to Robtarget

COMPAS **RRC**

Utilities

No-op (Ping)

Print Text

COMPAS RRC

Utilities

Wait Time (Delay)

Stop (Pause)

Stop Watch

COMPAS RRC

Utilities

Custom Instructions

COMPAS RRC

IO signals

Read analog

Set analog

Read digital

Set digital

Pulse digital

Read group signal

Set group signal



Work objects

```
# Define pick positions
frame_on_pick = Frame(Point(50, 50, 50), Vector(0, -1, 0), Vector(-1, 0, 0))
frame_on_place = Frame(Point(50, 50, 50), Vector(0, -1, 0), Vector(-1, 0, 0))

# Move to frame on pickup pallet (work object)
abb.send(rrc.SetWorkObject('ob_RRC_Brick_Pallet'))
abb.send_and_wait(rrc.MoveToFrame(frame_on_pick, speed, rrc.Zone.FINE))

# Move to frame on place (work object)
abb.send(rrc.SetWorkObject('ob_RRC_Build_Space'))
abb.send_and_wait(rrc.MoveToFrame(frame_on_place, speed, rrc.Zone.FINE))
```




Pick & place example

```
# Create a new brick
done = abb.send_and_wait(rrc.PulseDigital('doNewBrick',0.2))
# [...]

# Vacuum on
abb.send(rrc.SetDigital('doVacuumOn',1))

# Motion
abb.send(rrc.MoveToFrame(pre_place_position, speed, rrc.Zone.Z10))
abb.send(rrc.MoveToFrame(place_position, speed, rrc.Zone.FINE))

# Vacuum off
abb.send(rrc.SetDigital('doVacuumOn',0))
```

Next week

- Quiz -instead of coding assignment- due next week: Wed 5th May, 9AM.
- Ask for help if needed: Slack, Forum, Office Hours (Fridays, request via Slack)
- Next lecture:
 - Robot control exercise with real robot!

Thanks!

