

University
of Idaho

FANUC Python API

A robot story

By James Lasso



Github Repository

https://github.com/Uofl-CDACS/FANUC-Ethernet_IP_Drivers

- Install Process
- Link to documentation for class functions
- Some links Might be slightly dead because Garrett renamed the Uofl Organization

Getting Started

Prereqs

Prerequisites [↗](#)

This is an example of how to list things you need to use the software and how to install them.

- pip3

```
pip3 install pycomm3
```



- Install pycomm3
- pycomm allows us to use ethernet/IP



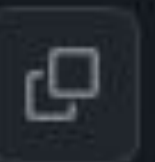
Getting Started

Installation

Installation

1. Clone the repo

```
git clone https://github.com/UniversityOfIdahoCDACS/FANUC-Ethernet_IP_Drivers.git
```



- Clone the repository
- YOU ARE READY TO ROBOT

Hello Robot

First Script

- !! CONNECT TO RIGHT NETWORK!!

```
# Imports
import sys
import time
import random
from robot_controller import robot

# Global Constants
#drive_path = '129.101.98.214' # Bill
drive_path = '129.101.98.215' # DJ
#drive_path = '129.101.98.244' # Larry
sleep_time = 0.5

▼ def main():
    """! Main program entry"""

    # Create new robot object
    crx10 = robot(drive_path)

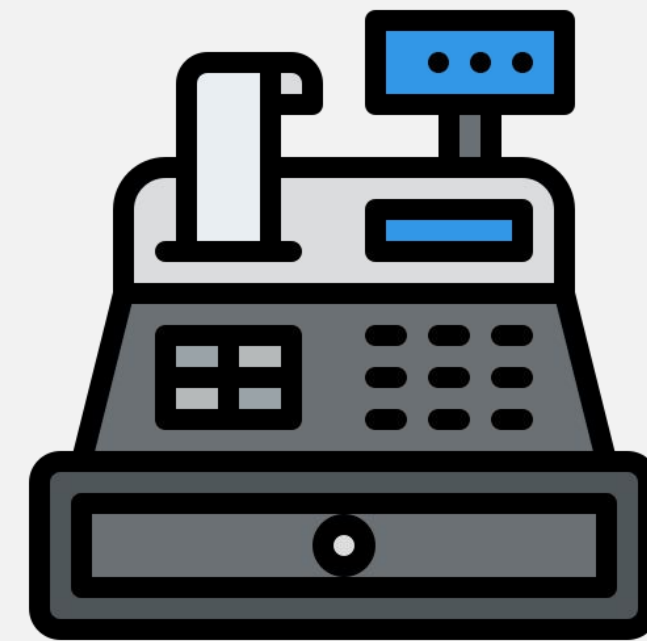
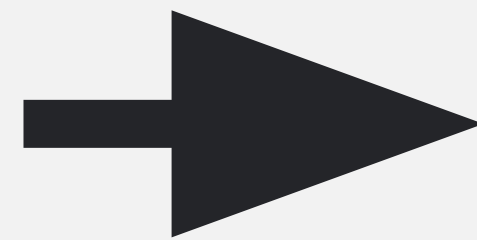
    # Set robot speed
    crx10.set_speed(400)
```

- CDA: tplink-robot
- SCOW: Im not sure

Basic Movement Usage

Set some registers to be written

```
crx10.set_pose(pose1)
```

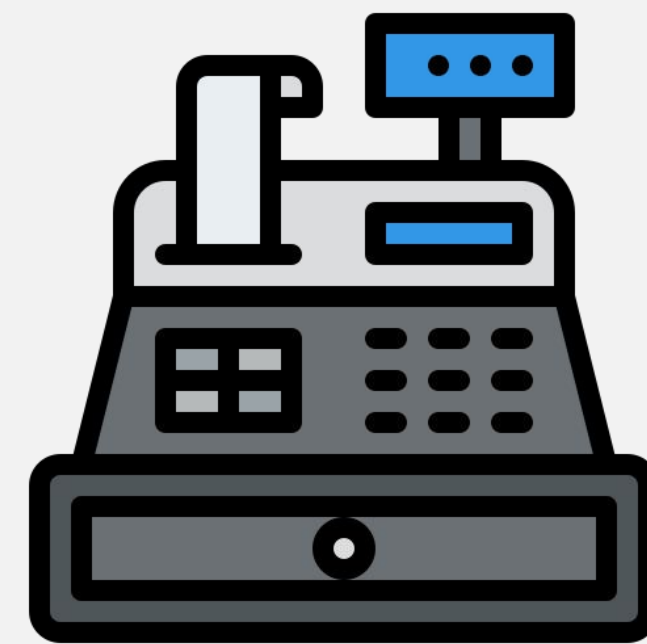
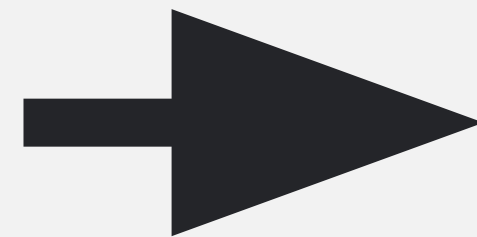


- External Register

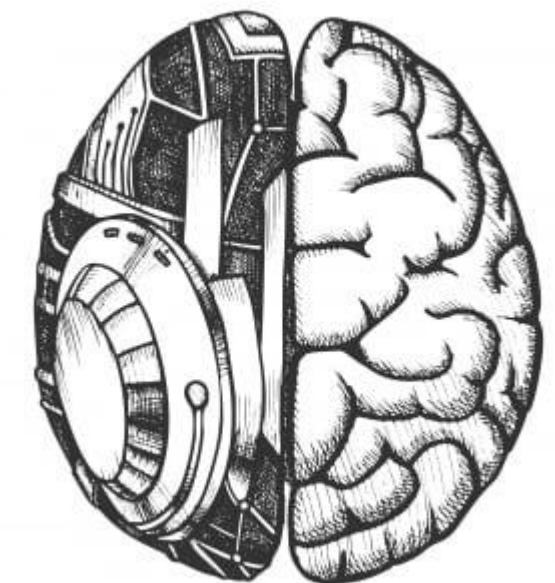
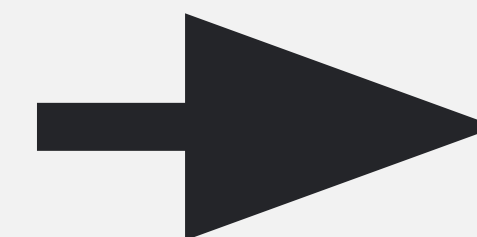
Basic Movement Usage

Use `start_robot()` function

```
# Sync bit and move robot  
crx10.start_robot()
```



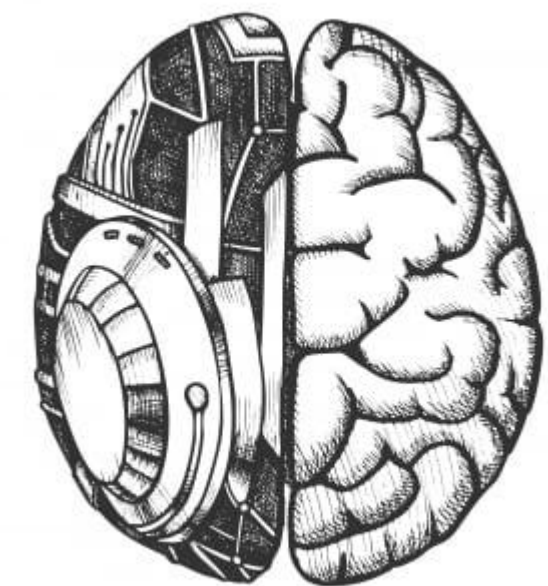
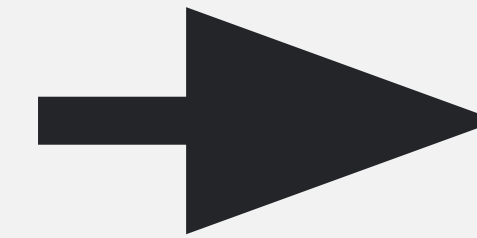
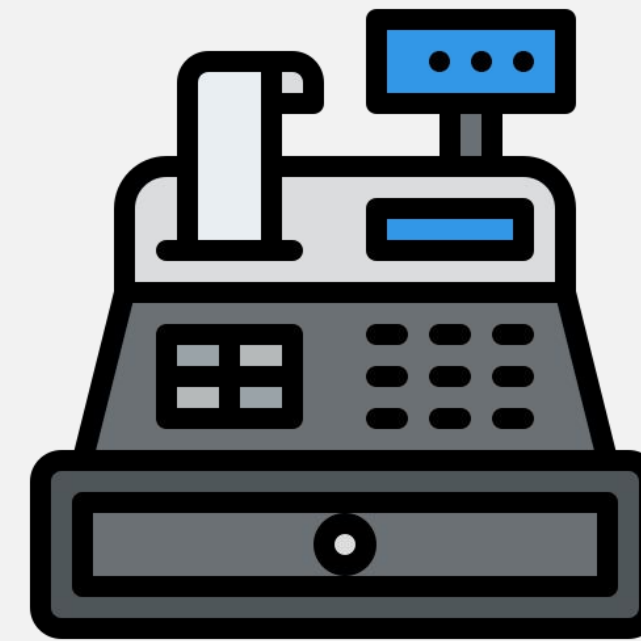
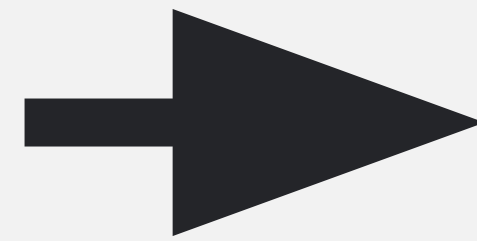
- External Register



- Internal Register

Basic Movement Usage

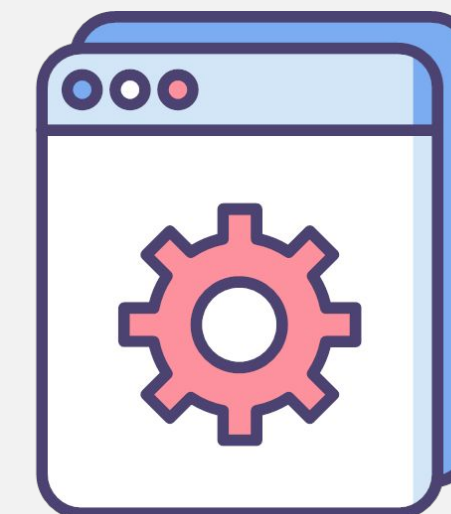
Use `start_robot()` function



- External Register

- Internal Register

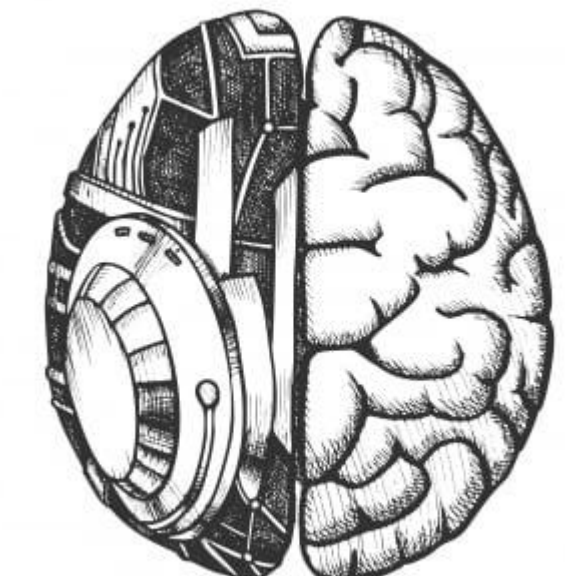
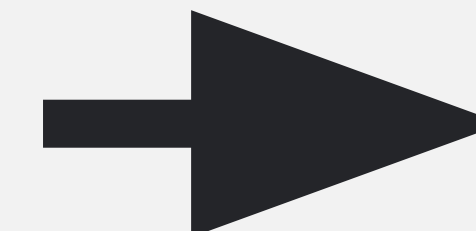
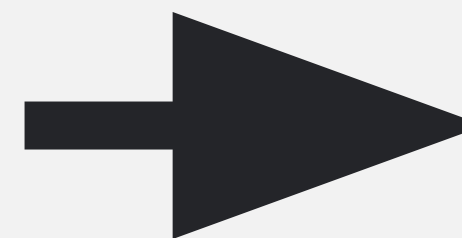
- Sync bit = 1?



- TP Program

Basic Movement Usage

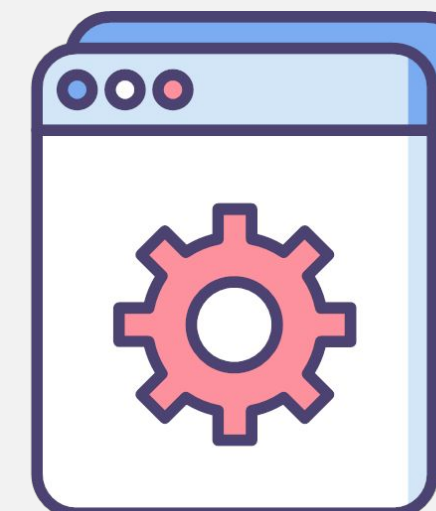
Use `start_robot()` function



- External Register

- Internal Register

- Sync bit = 1?

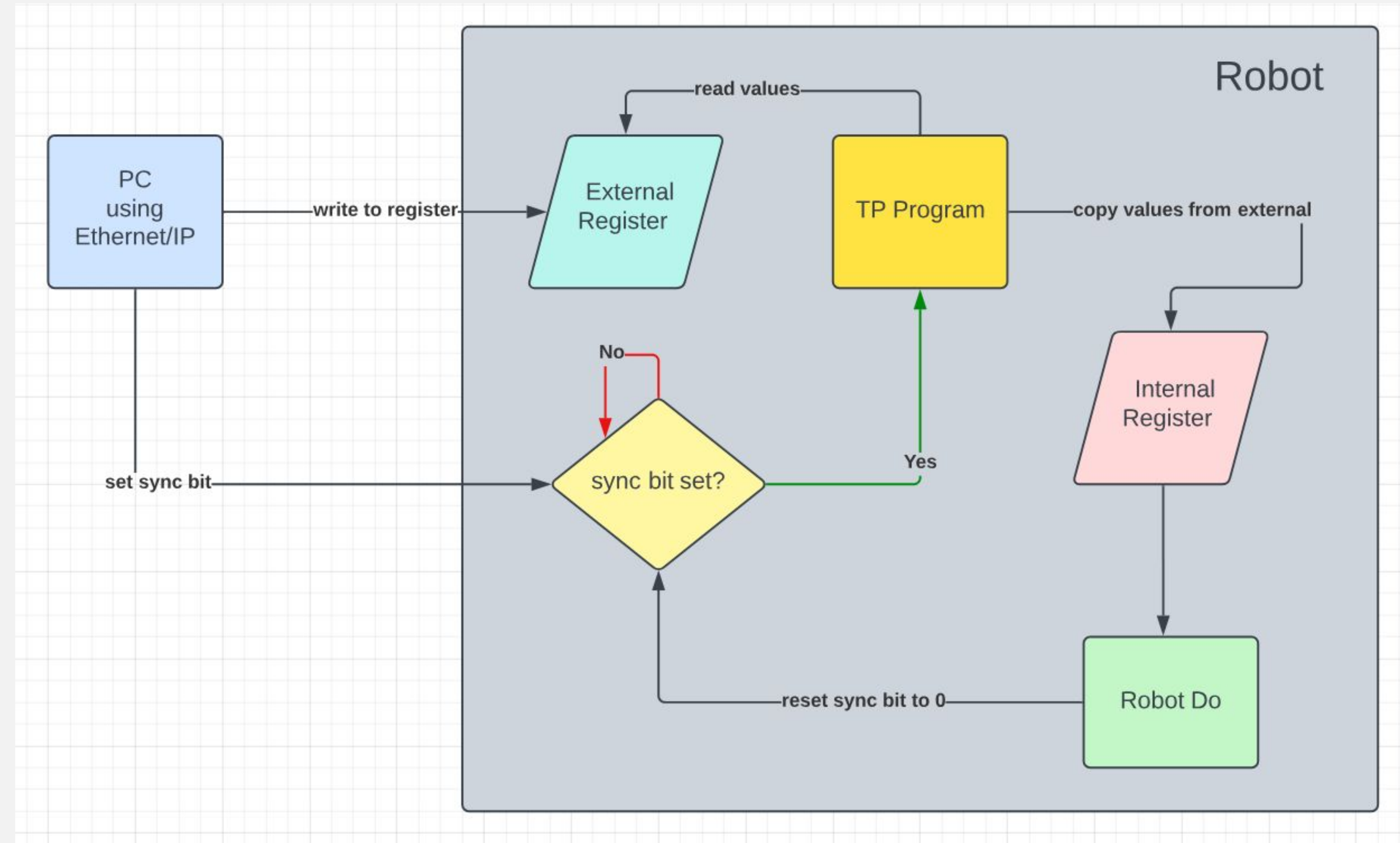


- TP Program

```
# Sync bit and move robot  
crx10.start_robot()
```

Block Diagram

Another Look



Basic Movement Usage

All together

```
# Move robot to home position and open gripper
crx10.set_joints_to_home_position()
# Sync bit and move robot
crx10.start_robot()
# Open gripper
crx10.gripper("open")

# Move to FIRST position (PREPARE TO PICK UP DICE)
pose1 = [14.000, 20.000, -45.000, -0.737, -46.000, 16.00]
crx10.set_pose(pose1)
# Sync bit and move robot
crx10.start_robot()
```


Basic Movement Usage

Same for cartesian

```
crx10.write_cartesian_coordinates(714, -170, 650)  
#crx10.write_cartesian_coordinates(714, -137, 650)  
# Execute move action  
crx10.start_robot()
```

Grippers/Proximity Sensors/Conveyor

Something Different

- Use different sync registers
- Do not need use start_robot() to use

Grippers

Modular

- Shunk and OnRobot grippers use different API functions
- These will eventually be in their own modules

Proximity Sensors

conveyor_proximity_sensor(sensor) sensor = 'left' or 'right'

- Will return a value of 1 or 0

Conveyor

`conveyor(command)`

- `command = string`
- `'forward'`
- `'reverse'`
- `'stop'`



Peek at the Docs

Please help

- All functions
- Their input values
- Return values
- Other fun stuff

Singularities

Blocked movements

- A configuration in which the robot cannot move to
- Wrist: Two robot wrist axes(4 and 6) line up with each other, joints try to spin instantly
- Shoulder: Center of robot wrist aligns with joint 1, causes joints 1 and 4 to try to spin instantaneously. Can also happen with joints 1 and 6
- Elbow: Center of robot wrist lies on the same plane as join 2 and 3
“reaching too far”
- More likely when doing inverse kinematics (end effector cartesian)

Useful Things to Remember

Movements

- Member variables are useful
- Joint positions start at index 2, below is joint 1

```
crx10.CurJointPosList[2]
```

- Try writing a helper script to capture pose positions
- If you are using cartesian coord movements find 0,0,0



Explore the Repo

One mans junk..

- Lots of examples
- Docs are a work in progress
- Feel free to email me with questions



Explore the Repo

One mans junk..

- Lots of examples
- Docs are a work in progress
- Feel free to email me with questions
- My email is jshovic@uidaho.edu

Explore the Repo

One mans junk..

- Lots of examples
- Docs are a work in progress
- Feel free to email me with questions
- My email is jshovic@uidaho.edu
- ASSIGNMENT: Repeat Lab 2 using the API