University of Essex

University of Essex

School of Electronic Engineering and Computer Science
University of Essex

Haptic-Controlled Robotic Arm Interface with Dobot Magician

The School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ, Essex, UK.

RUMI

Prerequisites and Setup

- Required Software:
  - Visual Studio (with Python & C++ extensions)
  - Python 3.13
  - DobotStudio
  - Haptic Device Software

- Project Folder:
  - Includes source code, DLLs, .bat file

- Environment Setup:
  - Add project folder path to environment variables

System Overview

- Core Files:

  - Master-Slave_Device.cpp: Haptic control, force feedback, pipe output

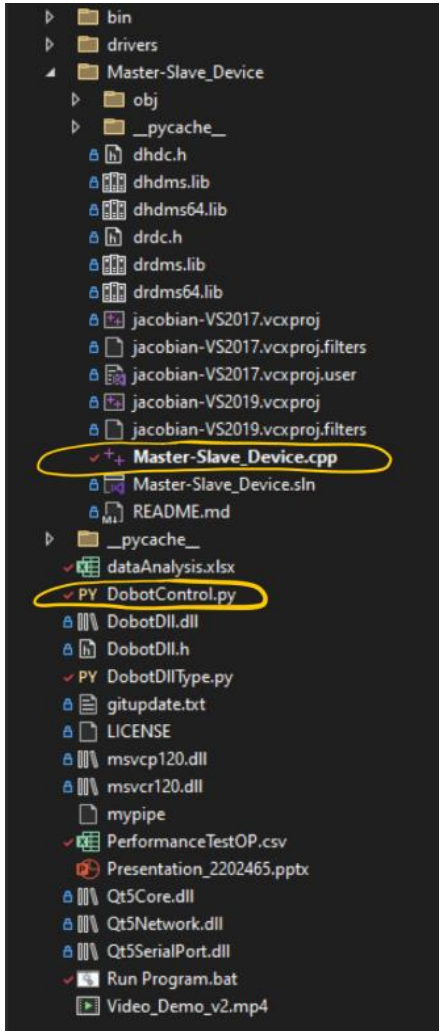
  The "Master-Slave_Device.cpp" file manages the connection and control of the haptic device and sends data to the Python script controlling the Dobot Magician (DobotControl.py).

  - DobotControl.py: Reads pipe data, controls Dobot

  "DobotControl.py" handles data processing from the haptic device and controls the Dobot.

- Data Flow:

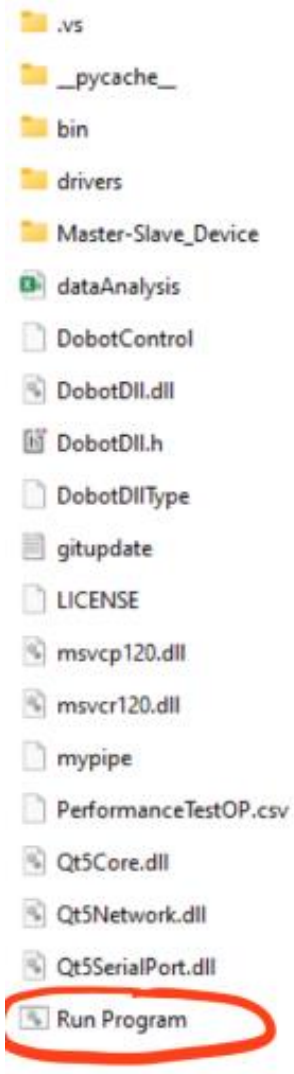
  Haptic Device → C++ → Pipe → Python → Dobot



Running the Program

1. Launch the .bat file

After the set up is complete, you can run the program by clicking the .bat-file in the project folder.



2. Two terminal windows will open:
   - **Haptic communication**
   - **Dobot control**

Adjusting Dobot Movements

- Python Control Command:
- dType.SetJOGCmd(api, 0, cmd, 0)

- **Command Selection:** Determines the appropriate command (`cmd`) for the robot based on the processed input values and button states.
  - `cmd = 1, 2` : Move along the X-axis (positive/negative).
  - `cmd = 3, 4` : Move along the Y-axis (positive/negative).
  - `cmd = 5, 6` : Move along the Z-axis (positive/negative).
  - `cmd = 7, 8` : Rotate the catheter (clockwise/counterclockwise).
  - `valueb = 1, 4` : Drive the catheter in or out using digital I/O pins.
  - `valueb = 5, 10` : Lock/unlock the robot's movement to allow for catheter manipulation without moving the arm.

```
# set coordinates for the dobot reading the valeus from the pipe.
if lock == 0:
    if valuex>0:
        cmd = 1
    elif valuex<-0:
        cmd = 2

    elif valuey>0:
        cmd = 3
    elif valuey<-0:
        cmd = 4
    elif valuez>0:
        cmd = 5
    elif valuez<-0:
        cmd = 6
    elif valueb == 8:
        cmd = 7
    elif valueb == 2:
        cmd = 8
    else:
        cmd = 0

#map the valocity of the arm according to the radius of the haptic devices input
vel = math.sqrt(pow(valuex,2)+pow(valuey,2)+pow(valuez,2))

# send the command to move the dobot
dType.SetJOGCmd(api, 0, cmd, 0) #Uncomment this for Lorans project
```

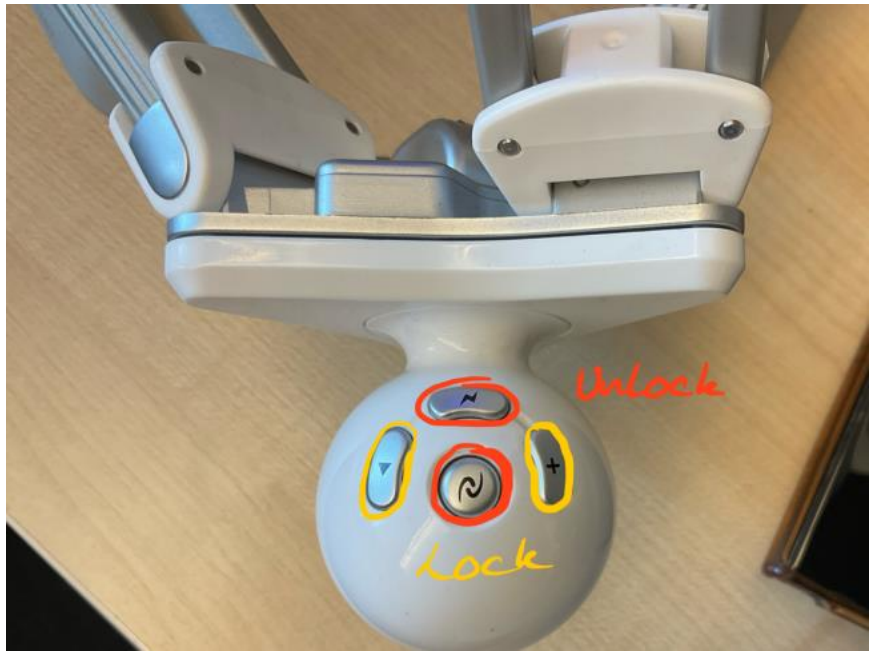
This is the part of the code that controls the Dobot movement.

For the Magnetic Guidewire's Application Test:

- 'cmd' controls direction
- Modified for X-axis only
- Movement scaled down for precision (For the application test is 10)

Guidewire Manipulation

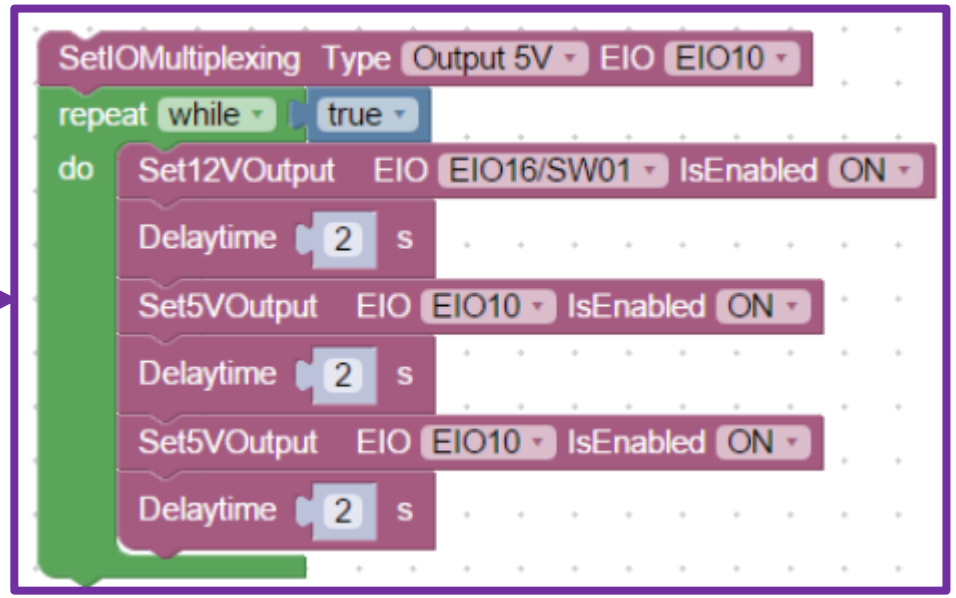
- Use buttons to move catheter in/out
- Direction may be reversed depending on USB port
- Press two buttons simultaneously to lock/unlock Dobot movement



Troubleshooting - Dobot

- **Issue:** Blue light, unresponsive Dobot
- **Fix:**

1. Open DobotStudio
2. Run Blockly program (EIO10 or EIO13)
3. Disconnect, close terminals
4. Relaunch the program



Troubleshooting - Haptic Device

- **Issue:** Device not detected
- **Fix:**

    1. Open 'Teneo' test tool
    2. Confirm detection
    3. Try different USB port
    4. Reboot or reinstall haptic software if needed