Bryan Walther

December 2018

Unified Robotics II Robot Arm

Setup and Known Issues

1. Setup
   1. Hardware  
      The operational box of the robot is a 20 inch semicircle centered on the middle of the robot (The output shaft of the base stepper). The robot has 180° of base rotation from it home switch clockwise.
      1. Prerequisites

5-6v DC power supply capable of delivering 5.5 Amps.

Webcam. Was tested using a Logitech 720p USB webcam. No other cameras were tested with the code.

Computer running the C# portion of the code. Was tested using a Fujitsu Lifebook T938, running 64 bit Windows 10. No other Computers were tested during this process.

Arduino Uno, or comparable microcontroller, containing ATmega328P controller. Was tested using a standard Arduino Uno. No other microcontroller was tested during this process.

* + 1. Working Environment

The robot is designed to be centered from on the working area (8.5”x11” Letter sized sheet of paper) at a distance of 12 inches from the center of the paper to the forward edge of the robot base. The page should be set up so the 11” are parallel to the front edge of the robot.

The camera is to be placed on the opposing side of the paper, is such a manner that the entire page is in frame. The C# code will crop to fit the page.  
Note: Ensure that the cord from the camera to the computer does not reach within the operational box of the robot to avoid issues.

The more accurate the lineup of the robot, camera, and paper, the better the pickup will be. Neither the robot, nor the C# code have a way of knowing the distance relative positions of any of those

* 1. Software
     1. Requirements

The robot requires the C# code (from the UR2-Robot-Arm-C-Sharp-Code repository) and the Arduino code (from the UR2-Robot-Arm-Arduino-Code repository).

* + 1. Running the Software

Ensure that the microcontroller (with the C++ code on it) and the camera are plugged into the computer. The code will function as long as the components are plugged in before the User tries to connect to the device.

Start the C# code, connect to the camera (from the drop down at the top), then open serial communication from the drop down towards the bottom.

At this point, the camera image and ROI should be visible in the UI. Select the Run Robot Checkbox to enable the robot and click Home to begin the process.

1. Known Issues
   1. Hardware Issues
      1. Stepper Strength

The 13Ncm steppers are at their limits with the current arm weight and shape weight. They can lift the arm from any position, but may struggle and loose steps as they raise the arm. To correct for this, the robot returns to its limit switch positions after every operation. This way, even if it does loose steps, it does not affect the future actions of the robot.

Having larger steppers may solve this issue, but that would require the arms to be larger to accommodate.

* + 1. Cable Management
       1. Length

The length of some of the cables are too short and can drag the middle motor controller to the left and right slightly as the arm rotates.

* + - 1. Routing

In a similar train, the wires are routed along the outside of the robot at the robot at the base. A better idea would have been to route the wires up through the center of the base. This also may alleviate the issue listed in 2.1.2.1.

* 1. Software Issues
     1. Serial
        1. Upon Opening

If the COM port is unavailable, the GUI will not list the port number. Closing and reopening the dropdown list will refresh it.

* + - 1. Timeout Exceptions

The serial monitoring thread will be sending a timeout exception every second that it is not receiving a message from the Arduino. These can be ignored and are simply to prevent the code from crashing due to no communications.

* + 1. Disconnections
       1. Arduino

If the Arduino becomes unplugged while the code is running, the software will crash. No safety measures have been implemented yet.

* + - 1. Camera

If the Arduino becomes unplugged while the code is running, the software will crash. No safety measures have been implemented yet.

* + 1. GUI
       1. Recalculate ROI Button

The effectiveness of this button is yet to be shown…

* + 1. Closing
       1. Thread Locking

When the form window is to be closed, all threads are locked until each can close safely. In rare occurrences, the serial monitoring thread can loop infinitely, freezing the window. The window is technically still responding to Windows, so the program must be force closed manually.