Robotic Arm Control Through Serial and WIFI Communication

Standard Operating Procedure

Jonathan Dean and John Zoodsma

# Project Purpose

The purpose of this program and device set is to control a robotic arm over WIFI. In order to do this, we are using a device to send remote commands to another device that will communicate serially to control the robotic arm.

# Setup

## Hardware

### The Arduino and the Raspberry Pi (intended for serial communication with the Arduino) should be mounted on the same surface as the robotic arm.

### Connect the Arduino to the robotic arm using the mounted motor driver shield.

### Power the Arduino using a 7-12V DC power supply.

### Power two Raspberry Pi’s using 5V DC power supplies.

#### A Raspberry Pi cannot supply enough power to the Arduino to be used as a power source for it, however we can program the Arduino using the Raspberry Pi over USB.

#### One Raspberry Pi will be used for serial communication (Serial) with the Arduino, and the other Raspberry Pi will be used for machine learning (Remote).

#### Both Raspberry Pi’s will communicate to each other and a server over WIFI

### Plug the Arduino into the Raspberry Pi (Serial) using USB.

#### **Begin Software Setup Instructions (Step B) at this point**

### Connect the mounted motor driver shield to a 12V DC Power supply to power the robotic arm.

### Make sure that the push button switch (blue) on the motor driver shield is pushed in.

#### The arm should be powered and ready to receive commands at this point.

## Software Setup

### Arduino Programming

#### Open the Arduino IDE.

#### Open file “controller\_1.ino”.

#### In the IDE, set the target board to be the Arduino Mega 2560.

#### Verify and program the device.

### Raspberry Pi Server

#### Open a new terminal instance and navigate to the file location of the file “PI\_Srvr.py”.

#### Open Python in the terminal.

#### Import the PI\_Srvr (server) class.

##### from PI\_Srvr import \*

#### Create a server variable with a port number passed as a parameter.

##### server = PI\_Srvr(10001)

###### This opens a server on port 10001

### Client Program (Remote)

#### Open a new terminal instance and navigate to the file location of the file “PI\_Cli.py”

#### Open Python in the terminal.

#### Import the PI\_Cli (client) class.

##### from PI\_Cli import \*

#### Create a client variable with an ip and a port number passed as parameters.

##### client = PI\_Cli(“127.0.0.1”, 10001)

###### This will attempt to connect to a server at IP address 127.0.0.1 (localhost) on port 10001.

#### The client should be connected to a server now.

### Client Program (Serial)

#### Open a new terminal instance and navigate to the file location of the file “PI\_SerCli.py”

#### Open Python in the terminal.

#### Import the PI\_SerCli (serial client) class.

##### from PI\_SerCli import \*

#### Create a serial client variable with an ip, a port number, and a baud rate passed as parameters.

##### ser\_client = PI\_Cli(“127.0.0.1”, 115200)

###### This will attempt to connect to a server at IP address 127.0.0.1 (localhost) on port 10001 and connect serially to the Arduino communicating at a baud rate of 115200.

#### The client should be connected to a server and to the Arduino now.

# Usage Instructions

## Commands

### The robotic arm can be sent commands on where to move the robotic arm specified by degree positions.

### Each command specifies a position and a servo to move.

### Servos are given values ‘a’ through ‘f’ labelled from bottom to top servo.

### To move the robot, a command has the format “<degrees><servo\_label>

#### Ex. A command that would move servo A to position 85° would be “85a”

### Multiple commands can be sent using commas and spaces.

#### Ex. “45a, 85b, 25c, 89d” etc.

# Device Description

A Raspberry Pi is used remotely to send commands to another Raspberry Pi. The second Raspberry Pi will relay the commands to an Arduino using serial communication. The Arduino has a program that will allow it to process the commands and to control the arm’s movement.

# Program Description

## Server Program

## Client Program (Remote)

## Client Program (Robotic Arm)

# Version

**Version : 0.0.3**

**Date: 11/04/2018**