Project Proposal

Human Interface for Robotic Control

SYSC 4907 - 4th Year Project

|  |  |
| --- | --- |
| Brandon To | 100874049 |
| Minh Mai | 100845949 |
| Yuzhou Liu | 100853392 |

Team Members:

Supervised by: Richard Dansereau

September 18th 2015

Carleton University

# Introduction

This document is a proposal for the SYSC 4907 fourth year project, under supervisor Dr. Richard Dansereau. The project will last from September 2015 to April 2016. The team consists of 3 members: Yuzhou Liu, Minh Mai, and Brandon To.

# Objective

The project aims to develop a robot arm system that mimics human movement and remote control over the internet with minimal latency and high accuracy.

# Motivation

The vision of this project is to develop a true embedded robotics platform that can serve as the foundations for developing robotic applications to replicate human movements. The motivation is to learn about firmware development for embedded microcontrollers. It allows the team to combine knowledge from both electrical and software disciplines.

Applications for this project range from manufacturing plant assembly lines, bomb diffusion, and remote surgery. Features such as remote control over internet and record/playback of user action were decided to be added to the platform to widen the scope of future development opportunities.

# Technical Overview

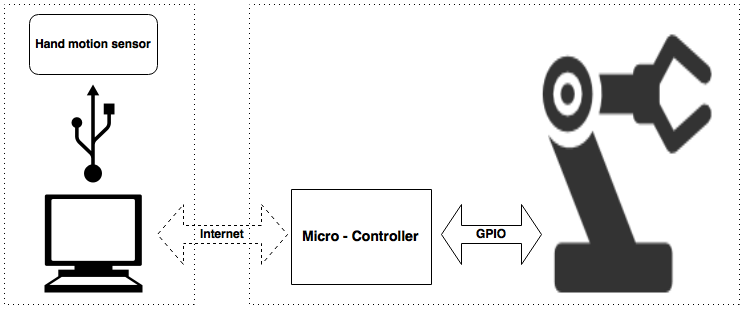


Figure 1: High Level System Architecture

In this project, the motions of the human hands are captured and sent to a computer. The program on the computer processes the data and sends it to the microcontroller via internet. Upon receiving the data, the microcontroller guides the robotic hand to replicate the user movements.

The hardware design consists of four main components: a hand motion sensor, a computer, a microcontroller, and a 3D printed robot hand (see Figure 1). The hand motion sensor used in this project is the Leap Motion. A Texas Instrument CC3200 SimpleLink Wi-Fi LaunchPad is chosen as a microcontroller. The hand will come from the InMoov open source robot project.

The software design consists of two component: the interface application on the computer and the firmware on the microcontroller. These components will be discussed below

## 3.1 Interface Application

The Leap Motion controller is a small USB peripheral device that provides accurate hand and finger tracking through the use of the sensor embedded in the device. A C++ program will be written which query the Leap device for relevant data, serialize the data, and stream it across the network. The decision of whether to use TCP or UDP for this has not yet been finalized. We would like to avoid using UDP if possible, to save ourselves the trouble of implementing our own reliability and network congestion protocol. Knowing this, we will initially develop our system under the assumption that we will be using TCP, and only switch to UDP if the transmission rate does not meet our expectations.

Here are some problems that we have identified that must be addressed in the communication across IP:

* The IP address of the microcontroller would have to be static and port forwarded on the router in order for an application to communicate with it from a different network.
* One group member who used the CC3200 WiFi shield (the WiFi module on our microcontroller) for a different project last year and mentioned that the shield does not work with enterprise WiFi.

To work around both of these limitations, we are planning on bringing our own router (set to bridging mode) to the poster fair demonstration.

## 3.2 Firmware

The microprocessor controls the robotic hand upon receiving the relevant data from the Interface Application. Firmware will be written to calculate the speed at which the joints are moving, and will configure the PWM accordingly in order to simulate that movement using the servo motors.

/\*TODO\*/ Brandon needs to reword

* The main challenge at this stage of the system will be the speed at which it can move will likely be much slower than the potential movement of our hand. We must handle the cases where the tracked hand moves faster than the servos can handle. As of now, our group believes that we should implement a queuing system in order to buffer the movement input before InMoov mimics it.
* The Leap Motion controller only tracks hand and finger movement. In order to incorporate arm movement, we must improvise. One idea is to only allow 1D arm movement, that is, only allow the arm to rotate along one axis. We can then use one previously untracked measurement (for example, the height of the hand) in order to dictate the movement of the entire arm. This is an overly simplified solution to a very complex problem due to hardware limitations, must it could be a first step.
* If we wanted to physically pick up items, we would need to include pressure sensors on the fingers in order to detect that we have actually come into contact with the object. This tells InMoov when to stop applying more pressure since the tracked hand will likely be out of sync with InMoov after InMoov comes into contact with an object.

/\*END TODO\*/

# Milestone and Timeline

The project is divided into 3 main milestones:

**Milestone 1:** Controlled software implementations

There are 2 pieces of software need to be written: the micro- controller firmware, the computers application.

**Milestone 2:** Interfacing the leap motion and the robot arm

The robot arm is required to mimic the human movement.

**Milestone 3**: Control-over-IP implementations

The control-over-IP software is written.

**Stretched Milestone 4:** Arm movements

Attach Timeline Yuzhou

# List of Special Components and Facilities

Listed below are the components that we need. These are subject to change at any given point:

[Yuzhou make table for this + include price]

* Hardware
  + InMoov open source 3D printed robot (hand, wrist and arm)
  + Leap Motion Controller
  + TI CC3200-LAUNCHXL
  + HS-311 Servo Motors
* Software
  + C++/Python for interfacing with the Leap Motion Controller
  + Berkeley Socket API for communication through IP
  + C on the microcontroller for developing the robotics platform