Team Number: 12

Project Members

Anthony Slas – aslas2

Mark Dabler – mdable2

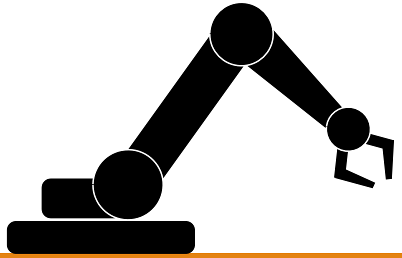
Robots Can Play Checkers Too

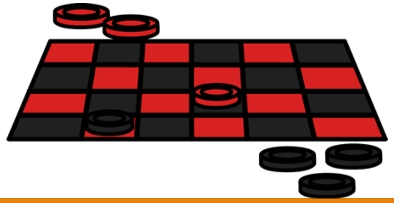
- Project Abstract -

This project will allow two robotic arms to go head to head in a game of Checkers. To do so, we will make use a few key project components; a game server to perform the game logic, video processing, and provide the user with a GUI to control the various parts of the game, a game camera to get an image of the game, two Arduino’s to control the robotic arms, and two robotic arms to perform physical game actions. The Arduino’s will communicate with the server via Wi-Fi.

Project Idea

The general idea is to have a computer system that will allow two robotic arms the ability to play against each other in a game of checkers. These two robotic will have the ability to reach the entire length and width of the game board to do so they will be driven by two Arduino Uno R3 microprocessors. The two Arduino Uno R3's will communicate with a game server (hosted on a desktop computer) via an external Wi-Fi chip. The Arduino's will gather the information needed for their robot's next move from the game server then translate the instructions into physical movements for the robot to perform game actions.





I/O Devices

Game Server	
Inputs	Outputs
● “Finished turn” signal from Arduino ● Image produced from the game camera.	● Next movement instruction to the correct Arduino. ● Game won, pause, stop, start signal to the Arduino. ● Other game options pertaining to the robotic arm.

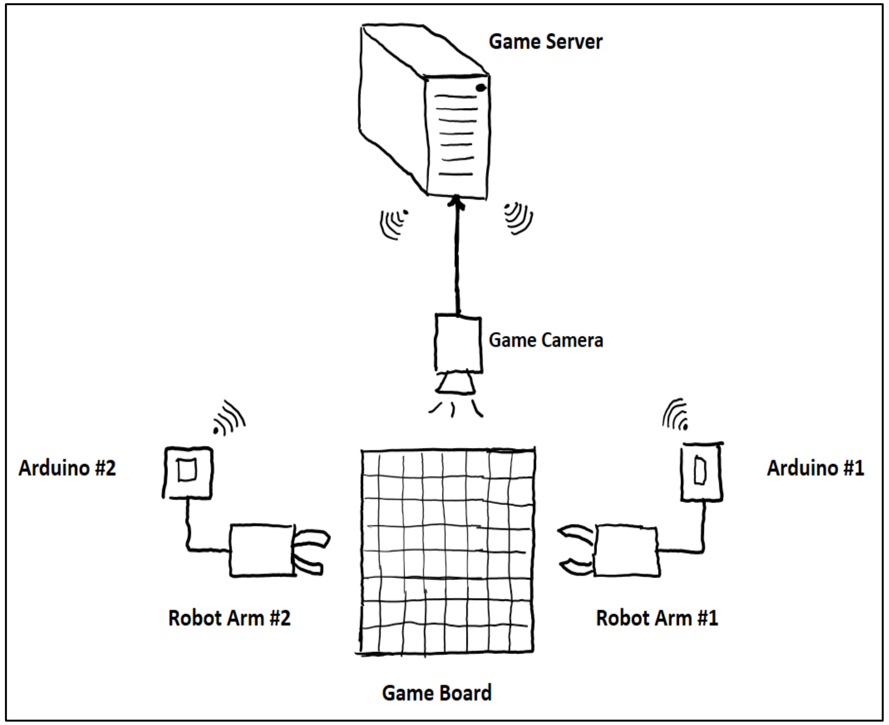
Game Camera	
Inputs	Outputs
● The image of the current game board.	● The image of the current game board.

I/O Devices (Cont.)

Arduino Uno R3 Microprocessor	
Inputs	**Outputs**
• Current move instruction from the game server. • Game won, pause, stop, start signal from the game server. • Other game options pertaining to the robotic arm	• "Finished turn" signal to the game server to the ESP8266 Wi-Fi card.

ESP8266 Wi-Fi Card	
Inputs	**Outputs**
• Move instructions from game server. • "Finished turn" signal from the Arduino.	• "Finished turn" signal to the game server.

Communication



There are two parts to our communication system. One for communication between the game server, and another for communication between the Wi-Fi card and the Arduino Microcontroller.

1. For communication between the game server and the Wi-Fi card we will use a UDP socket connection.

2. For communication between the Wi-Fi card and the Arduino we will use TX/RX serial communication.

Original Work

Most of this project will be done by us. We will be using other references only to help from the code base that will be used in this project. The game server will be written in Python, camera vision will utilize OpenCV, the game logic will be written in python which we will most likely use an already existing algorithm, and the Arduino/ESP8266 Wi-Fi card will be coded in C++.

What Worked

Some items that worked are as follows:

1. We were able to successfully establish communication between the Wi-Fi card and Arduino using TX/RX serial.
2. Our circuit design, although it was modified multiple times, worked. Our circuit is able to withstand powering everything up, as well as withstanding constant high current draws from each servo motor on the robot arm.

What Didn’t Work

Some items that didn’t work are as follows:

1. We were not able to create a reliable TCP connection to the game server with the libraries provided by ESP8266. As a work around the Wi-Fi card actually hosts a UDP server socket for communication.
2. 5V power via the Arduino GPIO was not able to supply enough power to our circuit, so we use an external 5V 1.5A power supply.

Roles

- Mark’s Roles:
  - OpenCV & computer vision calculations.
  - Checkers game algorithm.
  - Game server GUI.
  - Communication to Wi-Fi cards.

- Anthony’s Roles:
  - Circuit & game board design.
  - Mapping robot arm to game board positions.
  - Handling game server commands & communication to the game server.