Artificial Intelligence

Connect four

Installation Guide

Cuauhtémoc Suárez - A01206503
Mauricio Villanueva - A01202413
Arturo Rivera - A01202457

1.0 introduction

As final project of the class of artificial intelligence, our team decided to create a robot that will be able to play connect four. This robot will consist of 3 parts: A neural network capable of recognizing written digits to retrieve the user input of in which column the robot should put a chip; A robot capable of adding chips in a specific column on a connect four game, and a computer program capable of computing the best move the opponent should take in other to beat the human user in connect four.

In this document, it is explained all the materials, software and instruction you need to replicate this project and to be able to execute each module independently.

2.0 Software

There are some extra software you need to have installed in your computer, on the following list, a version of the program in which this project was tested is displayed next to the name. It might work with other versions but we have not tested yet

2.1 Required software

Arduino 1.6

- https://www.arduino.cc/en/Main/Software

Python 2.7

- https://www.python.org/downloads/

Python libraries:

- Opencv 3.2.0-dev
  - http://opencv.org/downloads.html
- numpy 1.11.2
  - http://www.scipy.org/scipylib/download.html
- pyserial 3.2.1
  - https://pypi.python.org/pypi/pyserial

*note: To install OpenCV, the team strongly recommend to follow this guide. http://www.pyimagesearch.com/2015/06/22/install-opencv-3-0-and-python-2-7-on-ubuntu/

2.2 Downloading repository

Once all the required software has been downloaded and installed, you need all the necessary files to run the program. First you need to clone the following repository:

https://github.com/franzmau/Connect4

All the commands that are shown on this guide must be executed inside the main folder of the repository.

2.3 Connect four

There are two versions of this game, one created in JAVA, and the other one that was programmed using python. Here are the instructions on how to run both.

2.3.1 JAVA connect four

Inside the main directory, there is a file named lab04.jar. This file contains the JAVA version of our connect four. To play, you need to execute the following command:

```
java -jar lab04.jar
```

To play, you need to enter a number from 0 to 6 that is the column you want to drop a chip.

2.3.2 Python connect four

To run the python version of the connect four, you first need to have an Arduino connected through the USB port because this program send a serial communication to it. If there is not Arduino connected, the program will crash.

First modify the file located at conecta_4_python/classes/SerialManager.py and replace on line 7 the port in which the Arduino is connected.

Inside the main directory, execute the following command to run the python version of the connect four:

```
python conecta_4_python/Main.py
```

To play, you need to enter a number from 1 to 7 that is the column you want to drop a chip.

2.4 ANN

For running (and training) the ANN separately from the other programs, you should have the following codes: network.py and mnist_loader.py. In addition, in the same folder there should be a folder called “data”, in there you must put the mnist.pkl.gz.

The structure is like this:

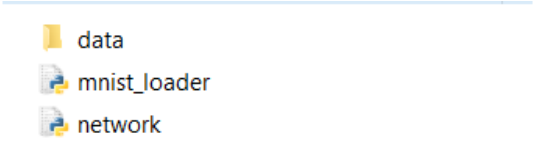


Figure 1. Folder Structure

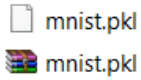


Figure 2. Inside data folder

Then, in the python command line, the next commands should be introduced:

```
>>> import mnist_loader
>>> training_data, validation_data, test_data =
mnist_loader.load_data_wrapper()
>>> import network
>>> network = network.Neuron_Network([784, 30, 10])
>>> network.Stochastic_Gradient_Descent(
training_data, 20, 10, 3.0, test_data = test_data)
```

In the command #4 the 784 means the number of neuron inputs, 30 is the number of neurons in the first layer (you can introduce more number after this value to create more layers), and 10 is the number of output neurons.

In the command #5 the 20 means the number of epochs (iterations for learning), 10 is the subsets of the MNIST data, and 3.0 is the learning rate.

2.5 Detecting numbers

For detecting numbers, firs you must be sure that you have at least one camera connected to your computer. Then you should edit the following file: “challenge/detect_number.py” on line 12 you must change the argument of the function to select the camera you want to use.

```
cap = cv2.VideoCapture(0)
```

If you have a laptop, 0 is almost always the integrated webcam, a 1 is an external USB camera. If you only have one camera connected to your computer, you should put a 0. In the function.

Once configured, in order to run the program, you should execute the following command:

```
python challenge/detec_number.py
```

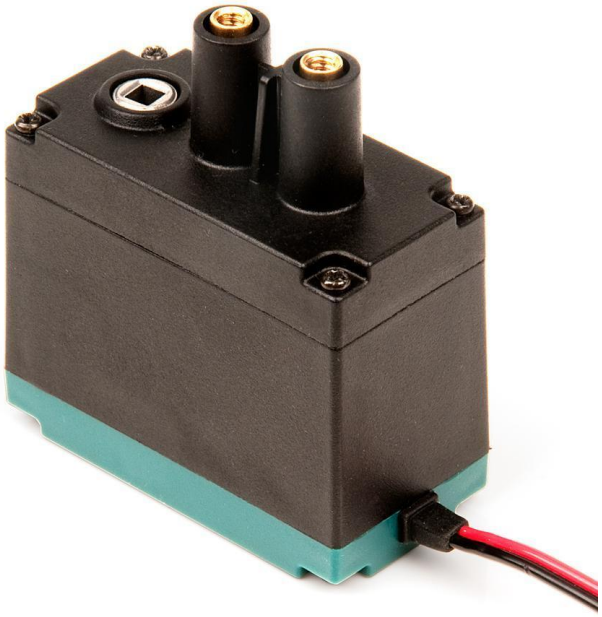
This program will use the camera to look for a number on a black and white board and will display the output on the terminal

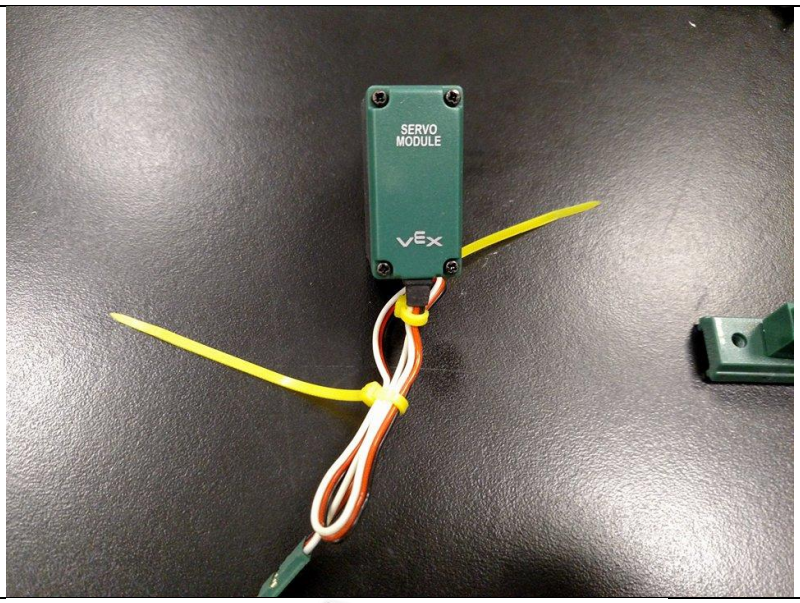
3.0 Hardware

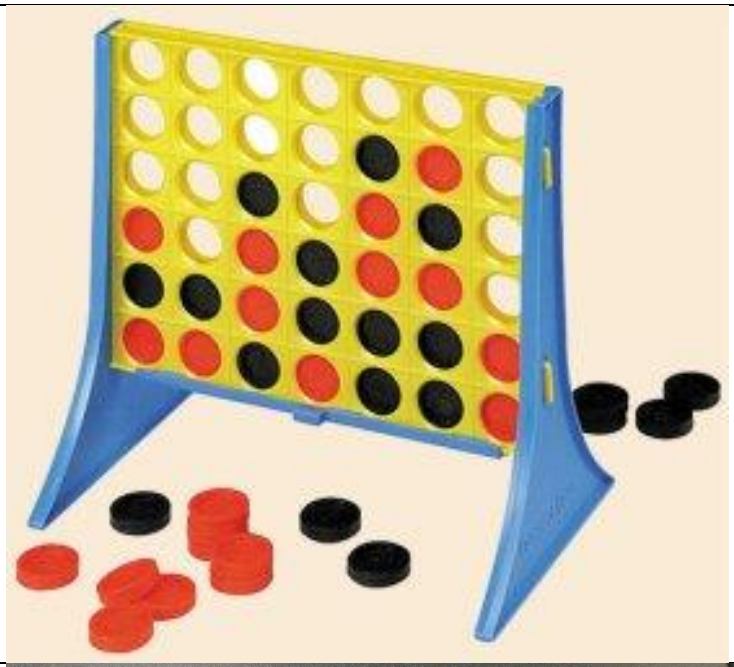
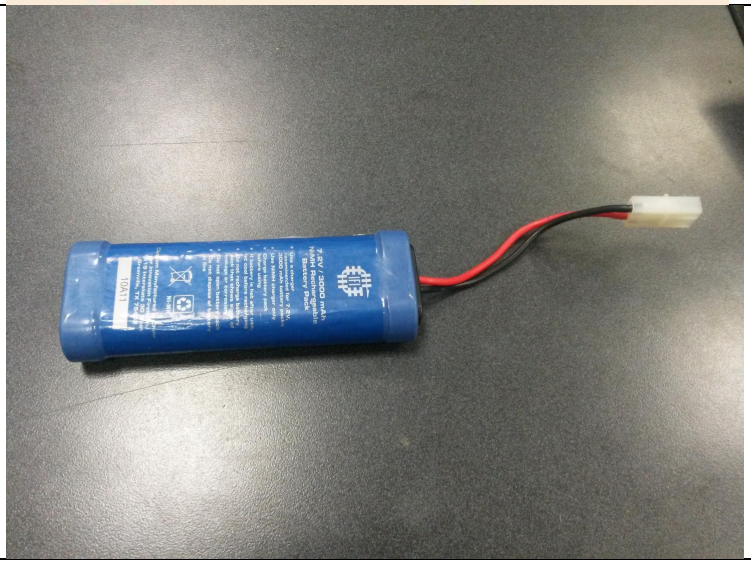
Because this game is intended to be played on a real connect four board, some hardware materials are needed. This is the way we create the infrastructure to the project but feel free to edit it or put suggestions.

3.1 Materials

The basic materials you will need to create the robot that will play connect four are the following:

Material	Image
1 vex motor model 383	

1 vex servo module	
Metal structure to support	
Wood coin infrastructure	

Connect four game		
7.1 Volts battery		

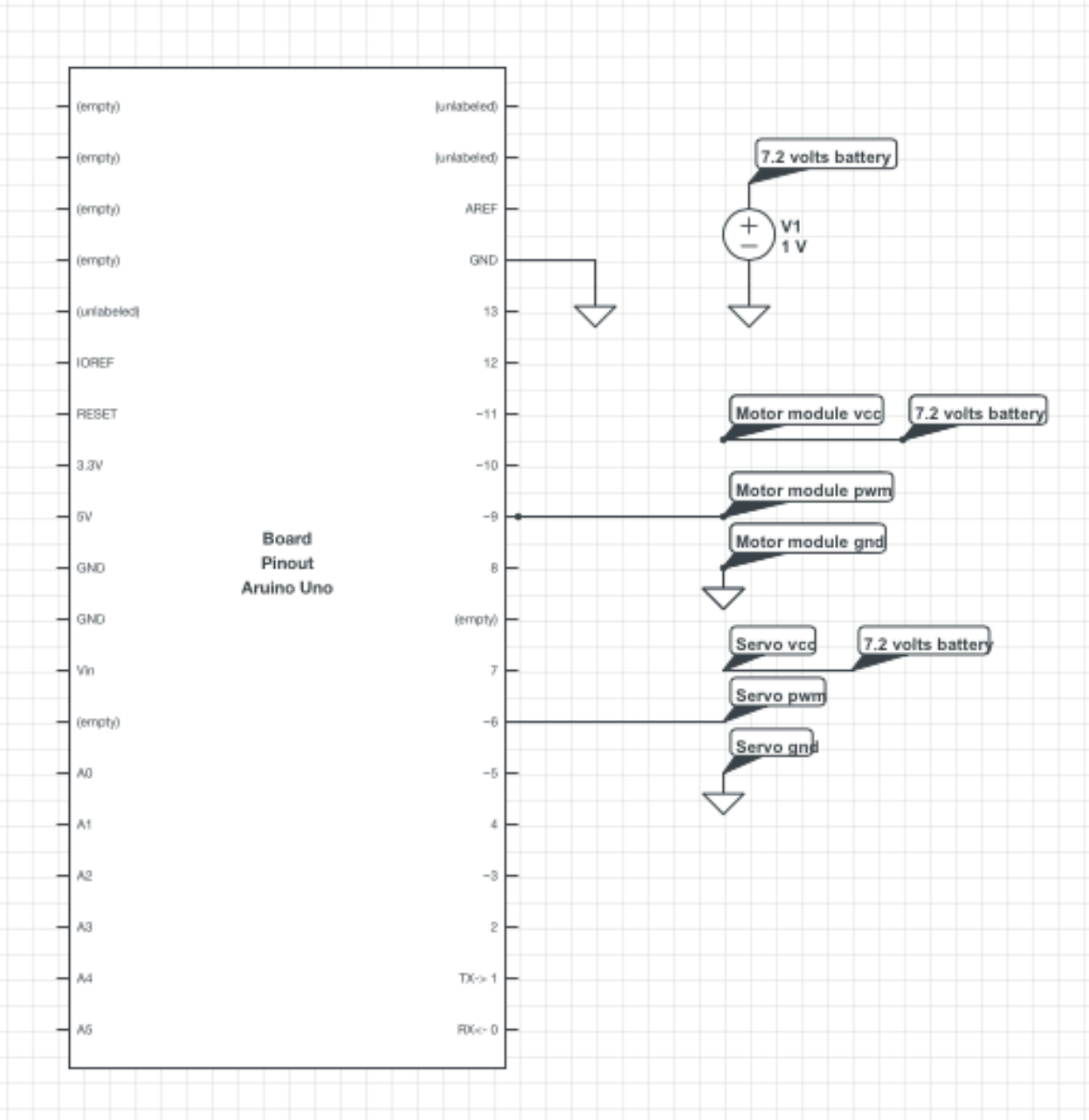
Arduino UNO	
Vex gears	

Vex motor controllers	
HP camera	

3.2 Schematic

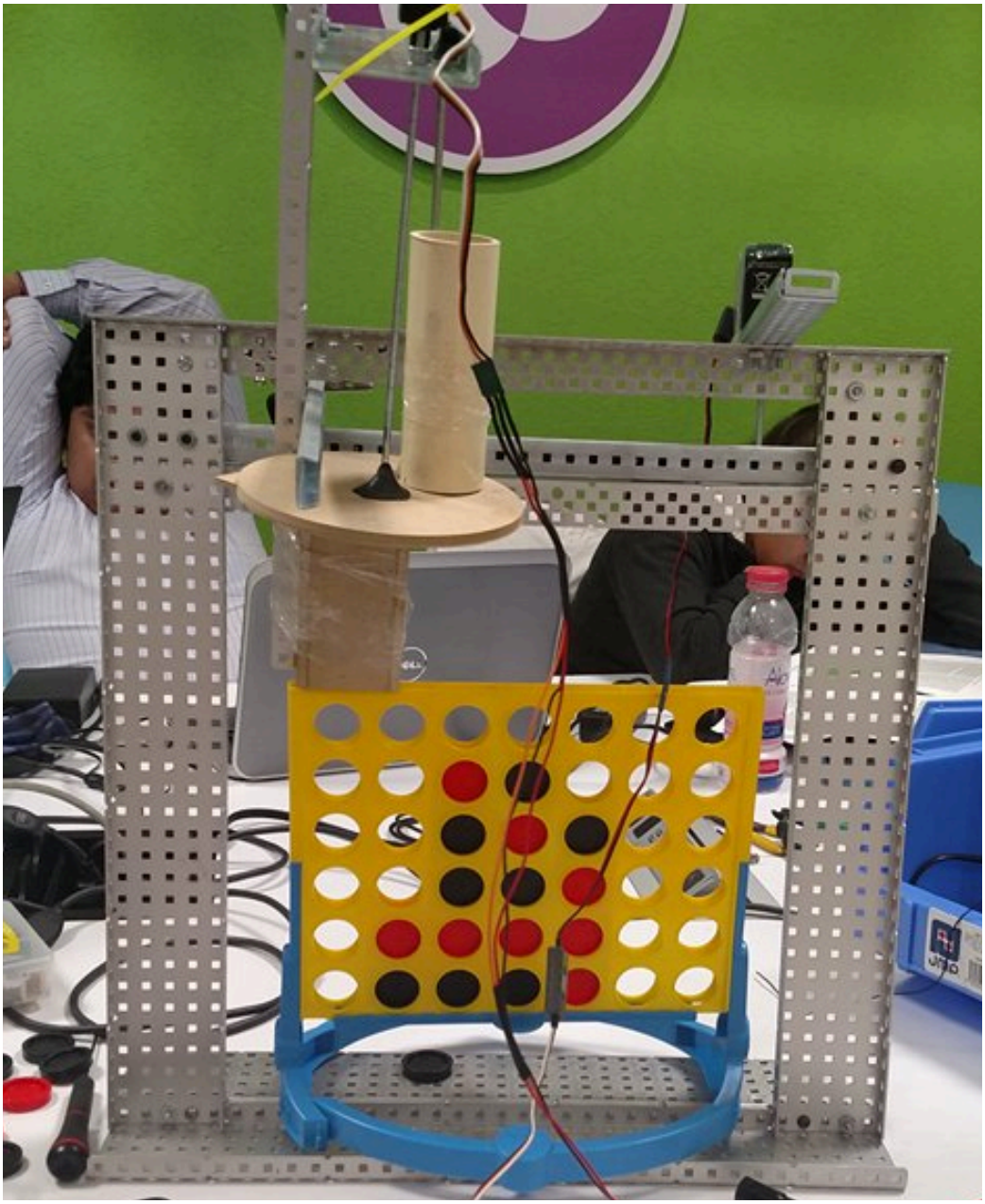
The circuit that makes the robot works is described in the following schematic, only to PWM signals are used from the Arduino uno to control the servo module and the motor, and both motors are connected to a

7.2volts battery.Also the Arduino was connected to a USB port of the computer:



3.3 Structure

At the end, we end up with the following structure that could move to a specific column with the motor module, and could drop the chip with the servo and the chip dispenser.



4 Comment's

If you have additional comments, or trouble following this guide or also if you have ideas on how to improve the project, share it with us via the following email address:

A01202457@itesm.mx

We will try to reply as soon as possible