

# Digital Control

## LABORATORY REPORT

---

# Flask and Ajax and Live sensor

---



**Amirkabir University of Technology**  
**(Tehran Polytechnic)**

Amirhossein Sheikholeslami

Mahboobeh Shakeri

Ahmadreza Nazari

Niloofar Raeesolsadati

Amin Razzaghi

## Contents

• Introduction .....	2
• Flask .....	2
• Ajax .....	3
• json and live sensor .....	4

## Introduction

There is a robot with position variable that can be changed manually or by setpoint we declare, we use Arduino as microprocessor (arduino uno board)

Actually This experiment is divided into two parts. First one is about reading and receiving the position of robot through the encoder-port.

The second task is reading the position of robot and then control it to a set-point that we declared. (we do this through the output-port in the device that specify by voltage.)

## Encoder

the initial concept is that we count pulses from the encoder with interrupt and when we have the number of pulses of encoder within the whole length of our path(means from zero position to the max position) ,we can read the position of robot in any x-position according to the number of all pulses . in this task we use encoder port that exist on the device as input-port of Arduino .

but the point is that we finally noticed that we can't get position of robot by encoder because the frequency of the encoder less than the frequency of the Arduino and ... so we can't do the task with this encoder.

```
from flask import Flask
app=Flask(__name__)

@app.route('/')
def myfunc():
    return "Hello World!"

if __name__=='__main__':
    app.run()
```

Figure 1: main structure of python code

As you can see in the figure we can change the url of web page by changing the inside of route function .

We can change it to anything that we want . we also can pass a variable to the that web page by using for example : `{variable}` inside the route and furthermore we can set the methods of that webpage for example to POST or GET .

For connecting our python code with html code , we must import `template` ,`render` above of our code

And the finally we can by importing `RPI.GPIO` send our command to the board like raspberry pi and then we can `digitalwrite` whatever pin we want

## Ajax

In this section our goal is getting the proper response from sever due to the user command without any refreshing the webpage. To do this we use Ajax .

Ajax actually is one of the subset of jquery that is one of the java script library itself .

And more specifier AJAX is the art of exchanging data with a server, and update parts of a web page - without reloading the whole page.

In first step we must add the script tag from `ajax.google` in the head section of our html code .

In this test and for understanding the abilities of Ajax we want to show the sum of 2 numbers that user enters .

We make a hyperlink with name `calculate` that whenever user clicks on it ,a function will call in the script tag and receive 2 numbers that was entered.

Note : we make this function in the script tag but we set the address host of this function to something else that in this code , we set to the `/addnumber`.

Actually the output of this function will pass to that address host .

As it was said above , with calling that function , address host is not the same before and another route function in python code will call that in there the output or request received will split and we can find that 2 numbers , the in inal step we return the `jsonify` of the summation of that 2 numbers

## json

at this section we want to send the temperature sensor to that localhost IP that we built by flask, ajax.

in first step we have to import arduinojson library above of our code in Arduino. after that and after testing that we connect to our hotspot that was explained in the previous experiments, we have to build a json object dynamically and make attribute for it like sensorname, node name and in next step we must add some nested array to our object like the value of sensor or the time that we receive the data.

then after reading the data by sensors we have to build a HTTPClient object and set the IP of the system having python code to it. we can see the IP through the our hotspot wifi and then we have append the 8080 that is the port number relating to the flask and our localhost and in its end we have to add the function name that we want to use in python that in this test is PostJson

after these steps we can post our data to that address. in the python code and in the postJson address host we can get the data and we can split it to our the structure that we built before

for showing better live, we take that data from postjson address and send it to a global variable. then in python code and in another address host we return the jsonify of content that global variable then in html code and in script tag we take that jsonify data by setting the same address host to the function.

and finally in this function we can split our data and give style to it that we show the data through a table.