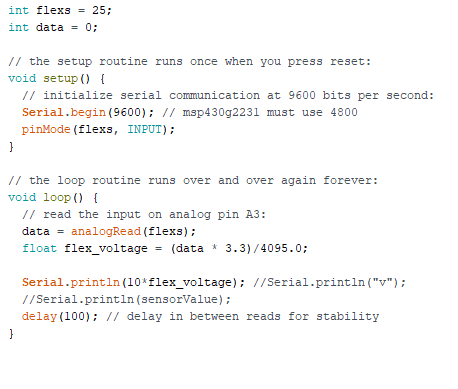
Flex sensor report

Member: Fahadh Mohamed - 62720,

Kuiyu Ding - 62716

Flex sensor :

Flex senor is a such type of sensor which measures the amount of bending or defection, it is made up with carbon surface on a plastic strip when this strip is bend or deflect then the resistance of this flex sensor is changed therefore it is also called bend sensor. It has two pins, P1 is connected to the ground pin and P2 is connected to data pin.

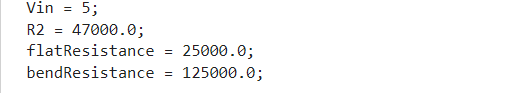


Physical and Energia part

4095 is the binary representation of the maximum input voltage(3.3v in this case). Tiva microcontroller has 12 bits, so 2^12 = 4096.

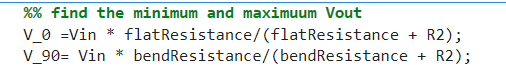
Matlab code:

Step 1 : Initially,



The Vin is set to 5V, R2 is set to 47K, the flat resistance at 0 degree is set to 25K, and the bend resistance at 90 degree is set to 125K.

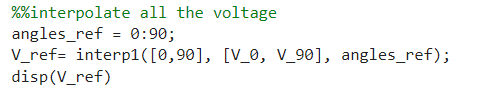
Step 2 :



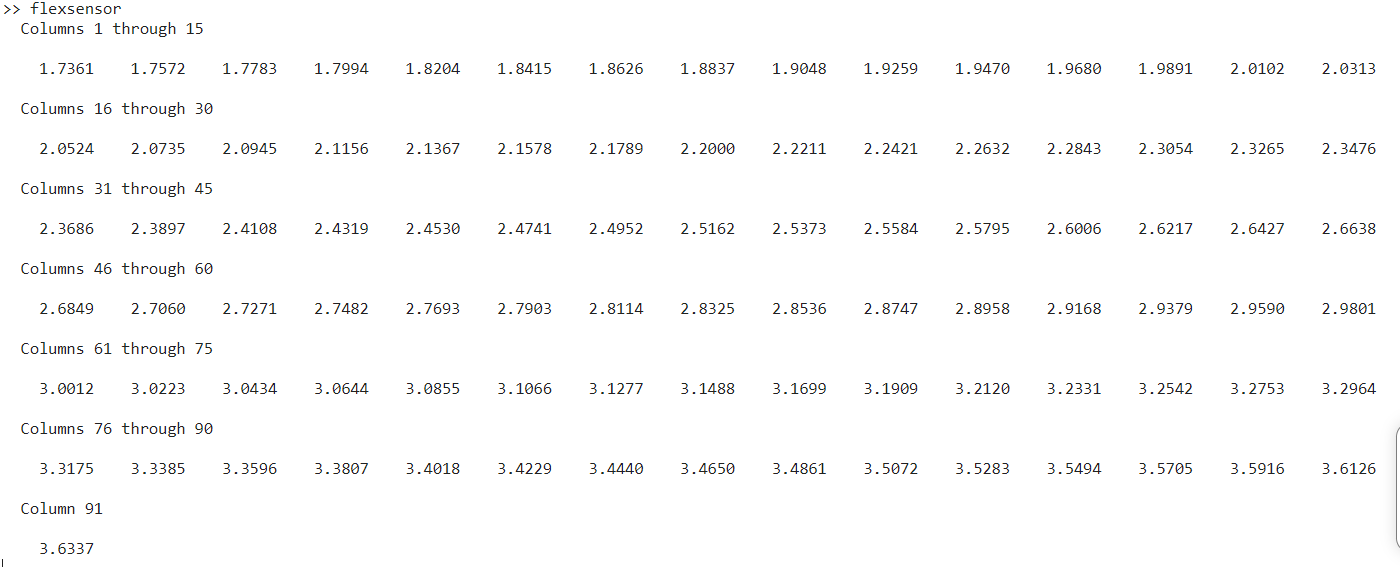
To calculate minimum voltage and maximum voltage, we use voltage divider formula, the formula is given above,

Now we have the value of Vmin and Vmax.

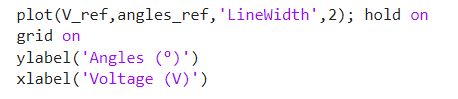
Step 3 :



Now, we are interpolating the voltage values from 0 degree and 90 degree by using interp1() function. We got the values like this,

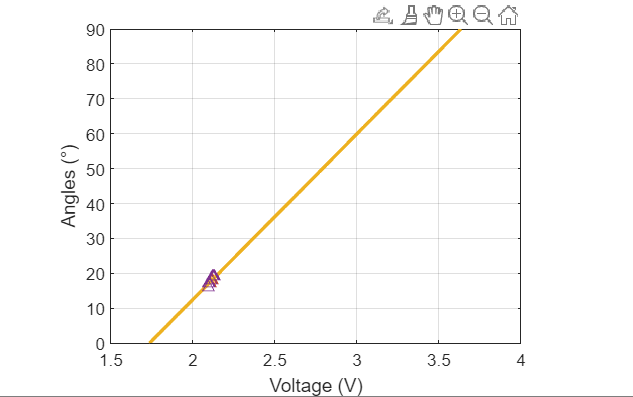


Step 4 :

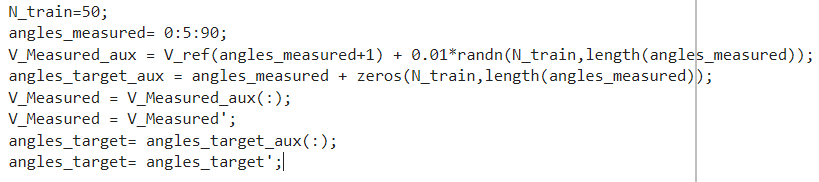


Now we plotted the graph using plot function between Angles ang Voltage.

We got the result relating the bent angles to the output voltage.



Step 5 :



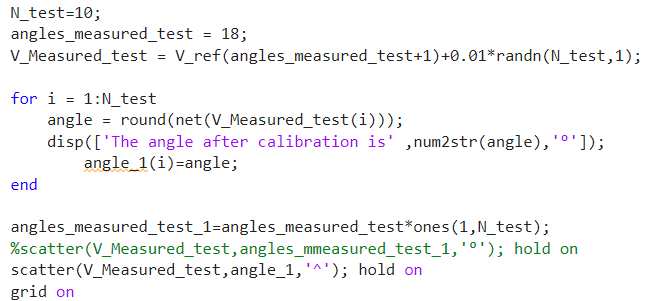
In this step, we considered N = 50 samples for each angle multiple of 5 degree, in the range between 0 and 90 degree, by adding a normally distributed error with standard deviation 0.01V , we used randn function to stimulate the normally distributed error.

Step 6 :

Picture 8

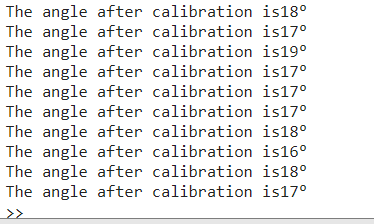
We used train() function, to train the shallow neural network H = 15, neurons in its hidden layer. It will receive voltage input data and present the calibrated angle as an output.

Step 7 :

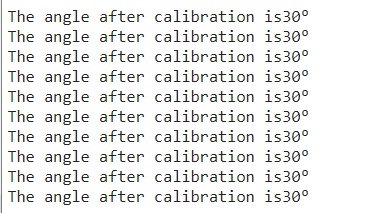


After obtaining the target model, it should be tested in a different dataset of samples, being independent from data used to train the NN. We tested N = 10 samples for the angle 18 degree with same error and standard deviation. We used round() function to the nearest integer to get the proper angular response.

Step 8 :

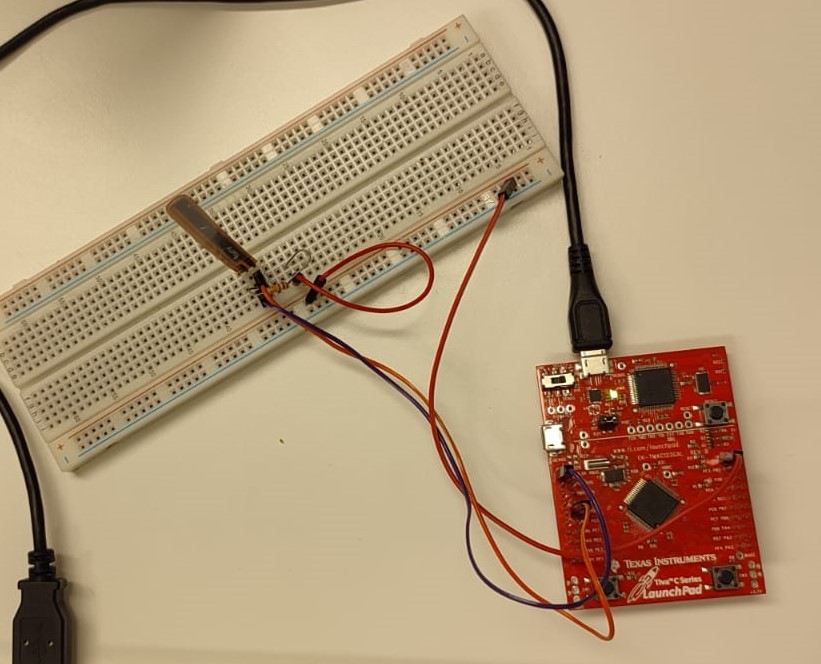


After testing, it gives approximately 18 degree.



we also tested for 30 degree, it gives accurate value of 30 degree.

**Train data and test part**



This photo represents the connection of flex sensor in bread board.

P1 is connected to data pin with 10K resistor. And P2 is connected to ground.

We found voltage using below formula,

V\_out = data\*3.3/ 4095

4095 is the binary representation of the maximum input voltage(3.3v in this case).

We downloaded data stream, it allows us to read the values in excel sheet.

We got the voltage values from 0 to 90 degree. And we trained the data.

D1 = readtable('/Users/coolbear/Desktop/total\_flexsensor\_value (5).xlsx','Range','A1:J70'); %Read the first 70 sets of data of the test data 0-90 degrees as the data for training the neural network

D2 = readtable('/Users/coolbear/Desktop/total\_flexsensor\_value (5).xlsx','Range','A71:J100');%Read the last 30 sets of data from 0-90 degrees of the test data as the data for testing and verifying the neural network

Matrice1=table2array(D1);

N\_train=70;

angles\_target\_aux = angles\_measured + zeros(N\_train,length(angles\_measured));%fill row vector angles as matrix vector

V\_Measured = reshape(Matrice1,1,700); %Convert Matrix Vector Measurement Data to Row Vectors

angles\_target= angles\_target\_aux(:);

angles\_target= angles\_target';

simplefit\_dataset=[V\_Measured;angles\_target]; %input value= V\_Measured, output value=angles\_target

net = fitnet(15); %Construct a function fitting neural network with one hidden layer of size 15.

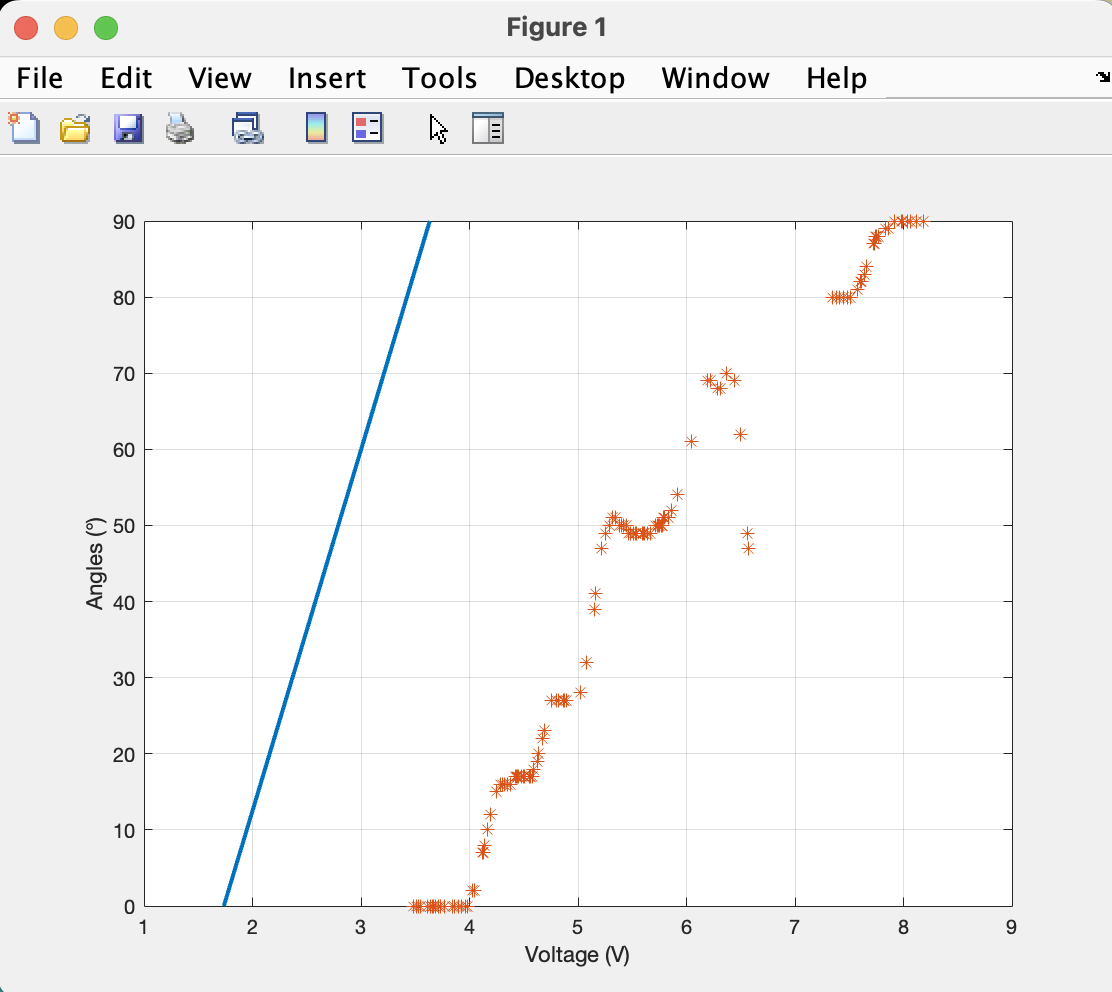
net= train(net,V\_Measured,angles\_target);

Matrice2 = table2array(D2);%Extract matrix data in D2

V\_Measured\_test = reshape(Matrice2,1,300); %Row vector becomes matrix vector

angle=round(net(V\_Measured\_test));%Input the test voltage into the tested neural network, get the angle output, and calculate its rounded result

scatter(V\_Measured\_test,angle,'\*'); hold on %Output the test results as a scatterplot

Test result:

After neural network training, the slope of the scatter plot is basically consistent with the ideal calculation value