Sign Language Interpreter Device

Introduction:

A lot of people among us do not have the ability to communicate with people by speaking. For these differently abled people there is a language called ASL (American Sign Language). But the problem is, even if some people know the signs, most of us do not know how to communicate to them with ASL. So, we proposed a design of a glove which can sense the Sign and interpret that in English, so that it becomes clear to people who do not understand sign language.

Instruments:

For detecting the motion of the wrist and fingers 6 flex sensors and 10 contact sensors are used. Flex sensors have a variable resistance which has a different value when it is bent compared to when it is straight. So it can be used to detect whether an individual finger is bent. Contact sensor senses whether different fingers touch each other. In understanding sign language it is vital to know which fingers are in contact. For that reason these many contact sensors are used to increase accuracy.

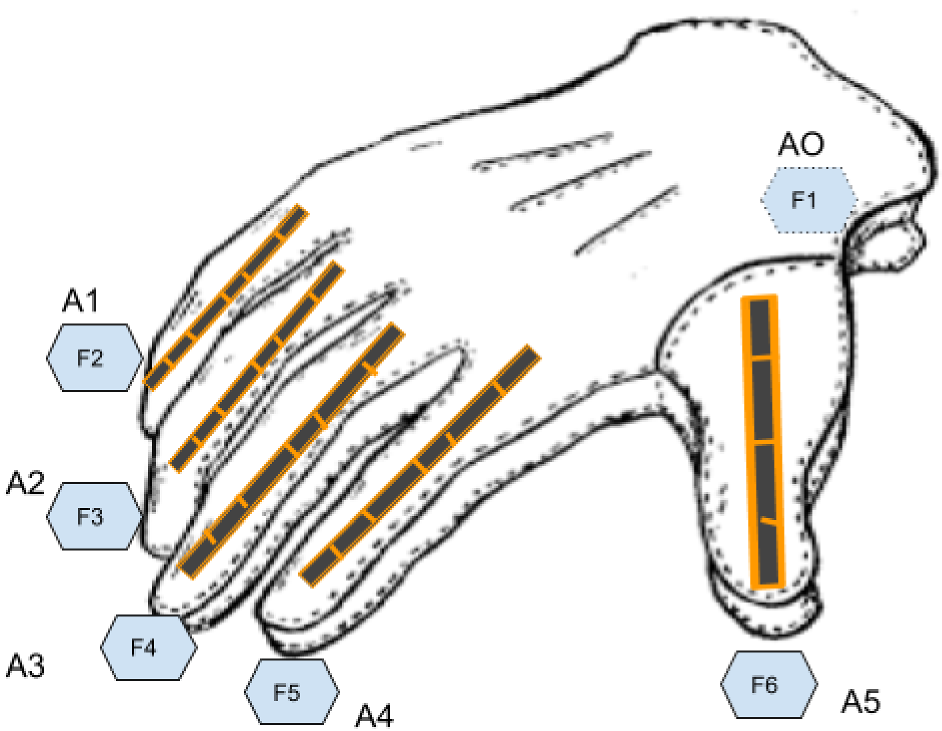

Figure: Back side of Glove with flex sensor and analog input notation

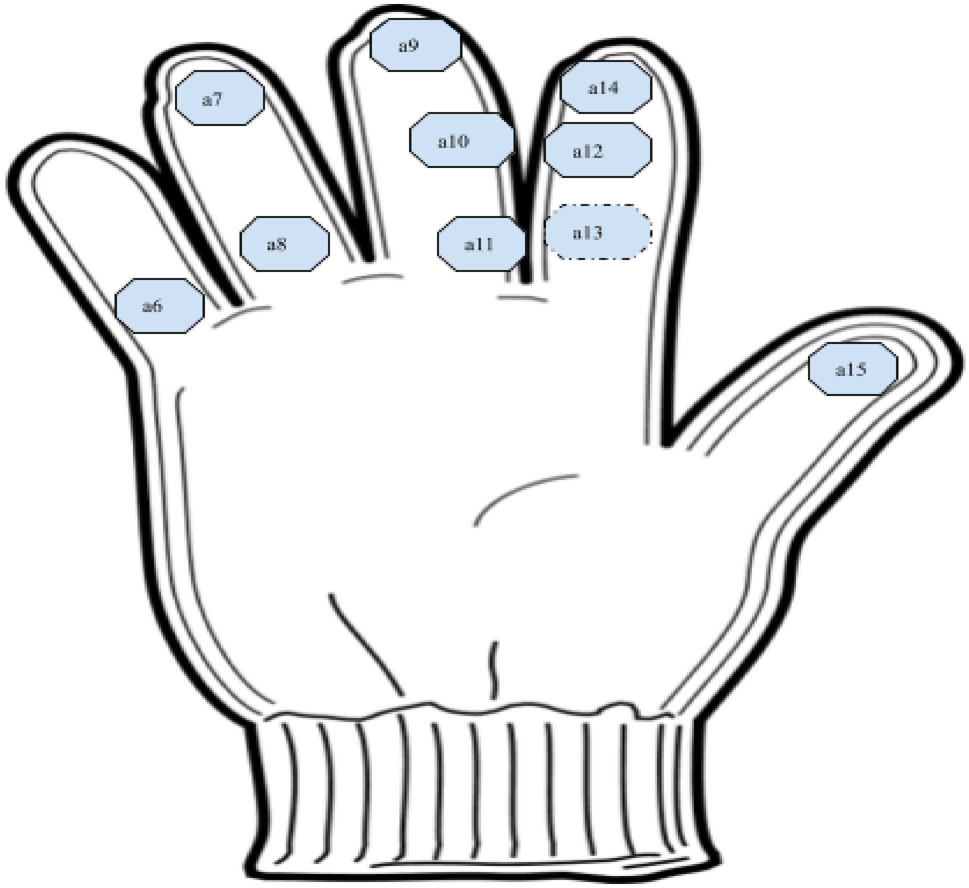

Figure: Front side of glove with contact sensors and analog input notation

How flex sensor works:

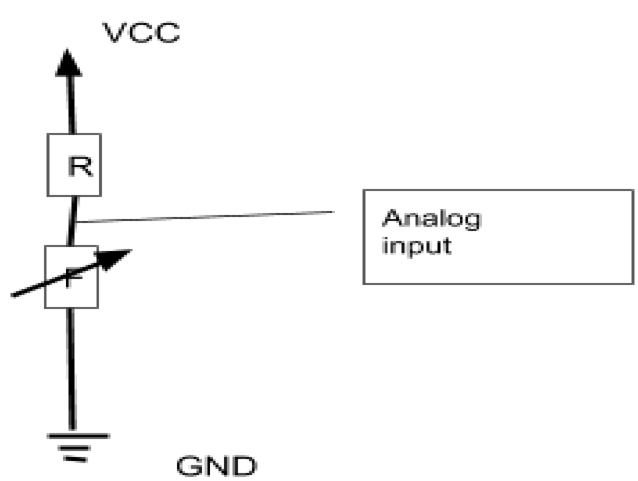

Figure: Connection of flex sensors to microcontroller

Flex sensors have a variable resistance inside of it which changes value according to how bent it is. They are connected in a way shown in the figure. It acts as a voltage divider. So the change in resistance is picked up by a microcontroller as a voltage analog input and data are processed accordingly.

How contact sensors detects contact:

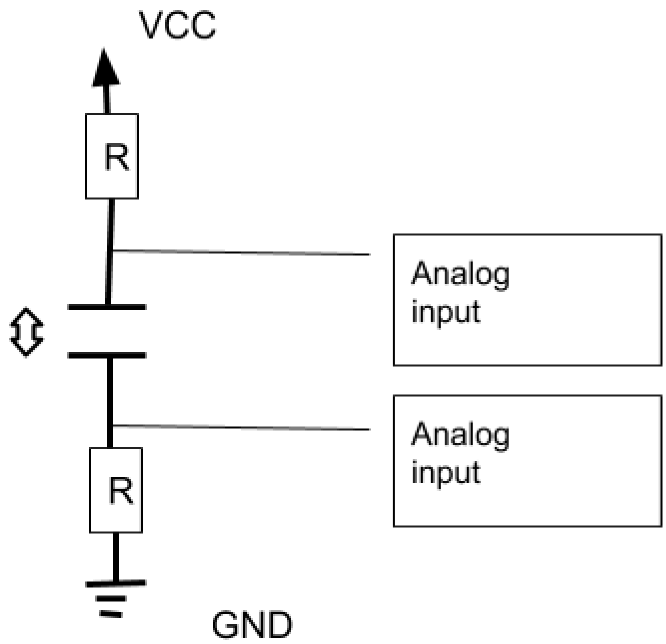

Figure: Connection of contact sensors to microcontroller

Contact sensors used here are circular iron wire spirals.Some sensors are connected to VCC by a resistance and others are connected to GND by some resistance. When the contact sensors do not touch each other, the voltages picked up by microcontrollers are either VCC or 0. But, when the contact sensors are in contact, current flows from VCC to GND and voltage drop takes place at the voltage divider which is picked up by the microcontroller.

Images of the device:

Figure: Backside of the glove

Figure: Front side of the glove

Description:

26 letters of english alphabet can be translated into sign language by the gestures of a single hand. These gestures required bending of different numbers of fingers and contact among them. With the 16 sensors installed on the glove, we can record the sensor values of the individual letters and try to learn the connection among the numbers.

A0 → Flex sensor in the wrist. Detects how much the wrist is bent
A1 → Flex sensor in the little finger. Detects how much it is bent.
A2 → Flex sensor in the ring finger. Detects how much it is bent.
A3 → Flex sensor in the middle finger. Detects how much it is bent.
A4 → Flex sensor in the index finger. Detects how much it is bent.
A5 → Flex sensor in the thumb finger. Detects how much it is bent.
A6 → Contact sensor in lower part of small finger. Connected to VCC.
A7 → Contact sensor in upper part of ring finger. Connected to VCC.
A8 → Contact sensor in lower part of ring finger. Connected to VCC.
A9 → Contact sensor in upper part of middle finger. Connected to VCC.
A10 → Contact sensor in middle part of middle finger. Connected to VCC.
A11 → Contact sensor in lower part of middle finger. Connected to VCC.
A12 → Contact sensor in middle part of index finger. Connected to GND.
A13 → Contact sensor in back part of index finger. Connected to GND.
A14 → Contact sensor in lower part of index finger. Connected to VCC.
A15 → Contact sensor in upper part of thumb. Connected to GND.

Figure: American Sign Language (ASL) Alphabet

Data collection:

For every letter nearly 50 samples are taken. These samples were cleaned up to remove any outliers. Mean value of every sensor of every letter is shown to understand the difference in between them.

	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
a:	404	880	649	848	769	671	1023	1023	1023	1023	512	1023	512	0	1023	0
b:	405	772	490	753	595	732	1023	1023	1023	1023	512	1023	512	0	1023	0
c:	401	832	617	832	717	709	1023	1023	1023	1023	512	1023	512	0	1023	0
d:	397	835	602	834	589	756	1023	1023	1023	512	1023	1023	0	0	1023	512
e:	393	860	625	823	733	787	1023	512	1023	1023	512	1023	512	0	1023	512
f:	407	756	479	744	694	743	1023	1023	1023	1023	1023	1023	0	0	512	512
g:	443	852	634	990	630	640	1023	1023	1023	1023	1023	1023	0	0	1023	0
h:	434	836	618	987	611	735	1023	1023	1023	1023	512	1023	512	0	1023	0
i:	415	747	621	813	738	750	1023	1023	1023	1023	512	1023	512	0	1023	0
k:	414	830	599	811	603	673	1023	1023	1023	1023	1023	512	0	0	1023	512
l:	409	847	623	829	582	610	1023	1023	1023	1023	1023	1023	0	0	1023	0
m:	397	839	596	796	722	736	512	1023	1023	1023	512	1023	512	0	1023	512
n:	380	812	596	792	723	694	1023	1023	512	1023	512	1023	512	0	1023	512
o:	375	816	568	971	681	738	1023	1023	1023	512	512	1023	512	0	1023	512
p:	501	781	568	777	592	697	1023	1023	1023	1023	1023	512	0	0	1023	512
q:	381	826	609	807	629	555	1023	1023	1023	1023	1023	1023	0	0	1023	0
r:	407	858	637	762	591	733	1023	1023	1023	1023	1023	512	0	512	1023	0
s:	385	831	645	822	755	805	1023	1023	1023	1023	512	1023	512	0	1023	0
t:	385	822	591	804	714	609	1023	1023	1023	1023	1023	512	0	0	1023	512
u:	394	845	568	768	581	722	1023	1023	1023	1023	512	1023	512	0	1023	0
v:	387	842	609	774	586	720	1023	1023	1023	1023	1023	1023	0	0	1023	0
w:	388	820	472	755	586	746	1023	1023	1023	1023	1023	1023	0	0	1023	0
x:	400	853	624	807	732	783	1023	1023	1023	1023	1023	1023	0	0	1023	0
y:	389	762	615	837	753	630	1023	1023	1023	1023	512	1023	512	0	1023	0

By looking at the data we can easily understand that the value of different sensors vary differently. The contact sensor has a range of 512 (from 0 to 512 or from 1023 to 512). On the other hand flex sensors have different ranges because of the nature of their internal resistor. Difference in letters like A and B can easily be understood by looking at the flex sensor inputs. But similar looking letters like U,V are tricky as the flex sensors show similar values. But the contact sensor differentiates them (A10,A12 shows that letter U requires contact of middle finger and index finger).

A lot of machine learning algorithms can be used to detect any new input by comparing that with a training set. But the nature of this project requires fast training and output. For that reason KNN (K- nearest neighbourhood) algorithm is used to detect the result.

But there is a problem with using this data as a training set. As the contact sensors have higher ranges of values compared to flex sensors, during finding the right answers the input of contact sensors will dominate the result. If by any chance mistakenly any contact sensors are not in contact, that error will be weighted highly and it will ignore the similarity in flex sensor values and give us the wrong output

For that reason, the inputs should be normalized to have a mean of 0 and e standard deviation of 1. The values will be weighted equally and it will be more likely to find correct output ,even if there is a mistake in input.

	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
a:	0.01	1.66	1.14	0.38	1.56	-0.60	0.21	0.21	0.21	0.30	-1.00	0.45	1.00	-0.21	0.21	-0.77
b:	0.05	-1.49	-2.16	-1.04	-1.00	0.41	0.21	0.21	0.21	0.30	-1.00	0.45	1.00	-0.21	0.21	-0.77
c:	-0.11	0.26	0.48	0.14	0.80	0.03	0.21	0.21	0.21	0.30	-1.00	0.45	1.00	-0.21	0.21	-0.77
d:	-0.26	0.35	0.17	0.17	-1.08	0.80	0.21	0.21	0.21	-3.32	1.00	0.45	-1.00	-0.21	0.21	1.29
e:	-0.42	1.07	0.65	0.01	1.03	1.31	0.21	-4.80	0.21	0.30	-1.00	0.45	1.00	-0.21	0.21	1.29
f:	0.12	-1.95	-2.39	-1.17	0.46	0.59	0.21	0.21	0.21	0.30	1.00	0.45	-1.00	-0.21	-4.80	1.29
g:	1.52	0.84	0.83	2.50	-0.48	-1.11	0.21	0.21	0.21	0.30	1.00	0.45	-1.00	-0.21	0.21	-0.77
h:	1.17	0.38	0.50	2.46	-0.76	0.46	0.21	0.21	0.21	0.30	-1.00	0.45	1.00	-0.21	0.21	-0.77
i:	0.44	-2.22	0.56	-0.14	1.11	0.70	0.21	0.21	0.21	0.30	-1.00	0.45	1.00	-0.21	0.21	-0.77
k:	0.40	0.20	0.10	-0.17	-0.88	-0.56	0.21	0.21	0.21	0.30	1.00	-2.24	-1.00	-0.21	0.21	1.29
l:	0.20	0.70	0.60	0.10	-1.19	-1.60	0.21	0.21	0.21	0.30	1.00	0.45	-1.00	-0.21	0.21	-0.77
m:	-0.26	0.46	0.04	-0.39	0.87	0.47	-4.80	0.21	0.21	0.30	-1.00	0.45	1.00	-0.21	0.21	1.29
n:	-0.93	-0.32	0.04	-0.45	0.89	-0.22	0.21	0.21	-4.80	0.30	-1.00	0.45	1.00	-0.21	0.21	1.29
o:	-1.12	-0.21	-0.54	2.22	0.27	0.51	0.21	0.21	0.21	-3.32	-1.00	0.45	1.00	-0.21	0.21	1.29
p:	3.78	-1.23	-0.54	-0.68	-1.04	-0.17	0.21	0.21	0.21	0.30	1.00	-2.24	-1.00	-0.21	0.21	1.29
q:	-0.89	0.08	0.31	-0.23	-0.50	-2.51	0.21	0.21	0.21	0.30	1.00	0.45	-1.00	-0.21	0.21	-0.77
r:	0.12	1.02	0.89	-0.90	-1.06	0.42	0.21	0.21	0.21	0.30	1.00	-2.24	-1.00	4.80	0.21	-0.77
s:	-0.73	0.23	1.06	0.00	1.36	1.61	0.21	0.21	0.21	0.30	-1.00	0.45	1.00	-0.21	0.21	-0.77
t:	-0.73	-0.03	-0.06	-0.27	0.75	-1.62	0.21	0.21	0.21	0.30	1.00	-2.24	-1.00	-0.21	0.21	1.29
u:	-0.38	0.64	-0.54	-0.81	-1.20	0.24	0.21	0.21	0.21	0.30	-1.00	0.45	1.00	-0.21	0.21	-0.77
v:	-0.65	0.55	0.31	-0.72	-1.13	0.21	0.21	0.21	0.21	0.30	1.00	0.45	-1.00	-0.21	0.21	-0.77
w:	-0.61	-0.09	-2.53	-1.01	-1.13	0.64	0.21	0.21	0.21	0.30	1.00	0.45	-1.00	-0.21	0.21	-0.77
x:	-0.15	0.87	0.62	-0.23	1.02	1.25	0.21	0.21	0.21	0.30	1.00	0.45	-1.00	-0.21	0.21	-0.77
y:	-0.58	-1.78	0.44	0.22	1.33	-1.27	0.21	0.21	0.21	0.30	-1.00	0.45	1.00	-0.21	0.21	-0.77

Figure: Normalized input data

Algorithm:

As a machine learning algorithm, KNN is chosen because of its relevance in this matter. In simple words, KNN plots the training data in a sample space having N(number of sensors) dimensions. When a new input is given, KNN plots that input in the sample space and finds its K nearest neighbours. Then it finds the neighbour who has the majority and accepts that as the answer. As the data shows variation between letters, this technique is meaningful and most intuitive and more likely to find the correct answer.

Result:

For evaluating the result, we took a testing dataset from the person who supplied the training data (as the gadget is most likely to be used by one person, so training data from that person matters the most). We also took some data from a different person to check the generality of the algorithm.

Letter	Accuracy(%)
A	95
B	92
C	94
D	97
E	96
F	94
G	94
H	95
I	96
K	96
L	97
M	92
N	91
O	94
P	93
Q	92
R	96
S	96
T	92
U	97
V	93
W	91
X	95
Y	96

The results show a high accuracy of more than 90%. Reason behind this high accuracy is the use of a large number of sensors and the use of machine learning. The slight amount of error is due to the fact that some test inputs were taken by different people wearing that glove. So the gestures varied as a different person gave the input. Moreover, some input was intentionally taken in a hurry which resulted in faulty contact sensor values. So accuracy can be increased further by taking the input more carefully.

Link of data and codes:

https://github.com/ishtiakm/sign_language_interpreter

Conclusion:

By using this gadget, a person can easily communicate with another person. More improvement can be done by training the glove to recognize words as well. The output can also be converted into audio. So, it will help a person to carry on a conversion in a much easier way.