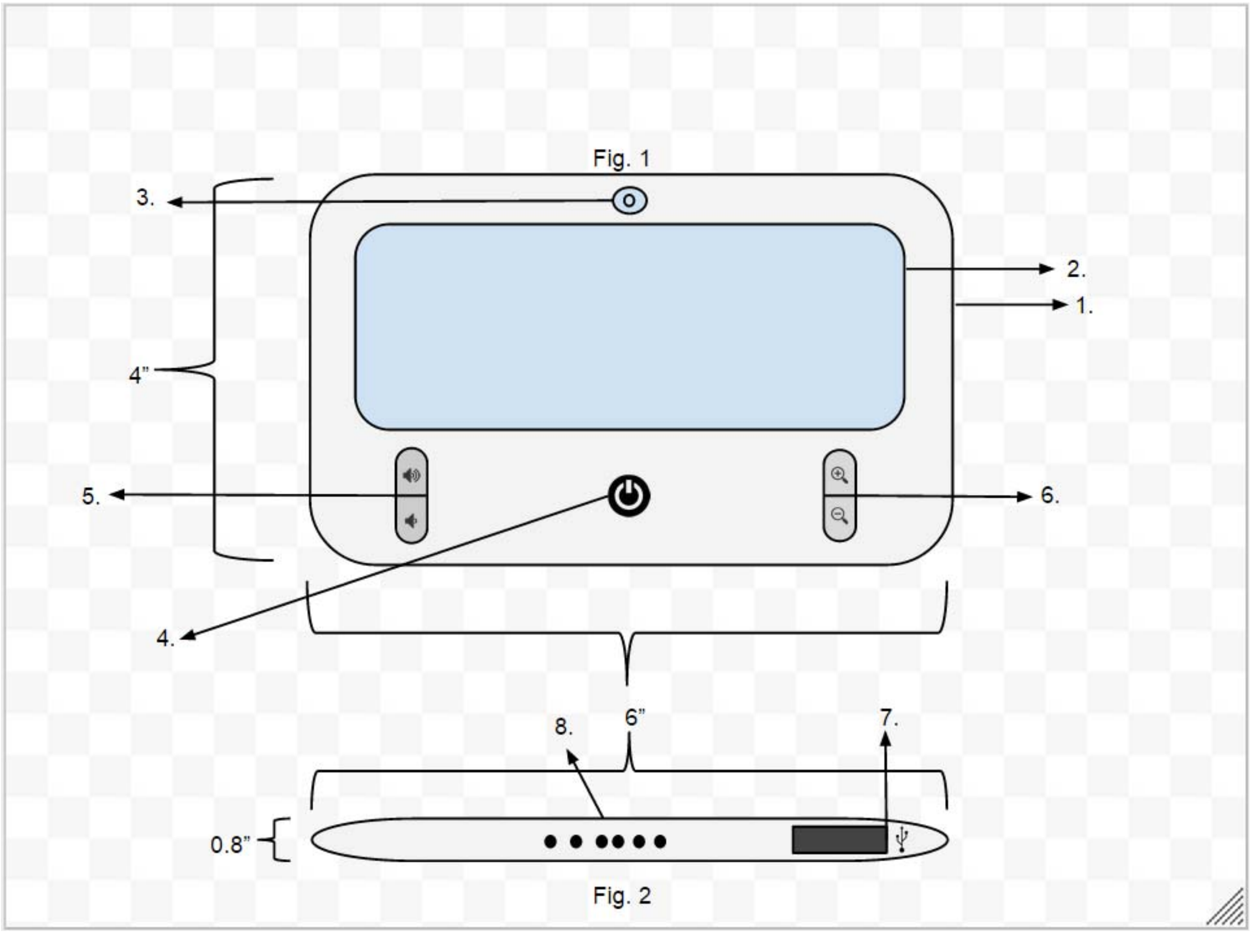
Team Name: Slate 8

Team Members: Yonatan Yilma, Marcos Celestino Carvalho Junior, Sarad Dhungel, Claude Ndzami, Renika Montgomery, Reginald Etienne.

Final Solution Design



1. **Design's Description:**

1.1. External Features:

Each team members design had great attributes, however, this design was chosen by the team because it presents the best attributes to develop a device that attends all the requisites established previously. Some attributes were analyzed such as weight, power, battery, display, camera, buttons, among others. The main device's features consists in a portable device that will perform the requisites. Therefore, as result, we have the external features' details:

Attributes	**Details**
Weight	15 ounces (425.25 grams)
Height	4 inches (10.16 centimeters)
Width	6 inches (15.24 centimeters)
Depth	0.8 inches (2.03 centimeters)
Display	2 inches (5.08 centimeters) by 4.75 inches (12.06 centimeters) 5.15 inches (diagonal) led screen 960-by-640 resolution at 326 ppi
Camera	5-megapixel camera embedded in device
Video	720p video recording (30 fps or 60 fps)
Power	AC converter and USB cable On the bottom right of the device
Battery	Built-in rechargeable lithium-ion battery Working time: up to 5 hours
Speakers	On the bottom middle of the device
Buttons	1 button for turn On and off 2 buttons for volume up and down 2 buttons for zoom up and down (to focus)

1.2. Internal Features:

The internal features of our device uses an Intel Galileo board. The description of the board is based on the following areas: physical characteristics, communication, processor features, and storage option.

a. Physical Characteristics

- 10 cm long and 7 cm wide with the USB connectors, UART jack, Ethernet connector, and power jack extending beyond the former dimension
- Four screw holes allow the board to be attached to a surface or case
- Standard 10-pin JTAG header for debugging
- Reset button to reset the sketch and any attached shields

b. Communication

- 10/100 Mb Ethernet RJ45 port
- USB 2.0 Client port

- B 2.0 Host port
- RS-232 UART port and 3.5mm jack
- Mini PCI Express (mPCIe) slot with USB-2.0 Host support

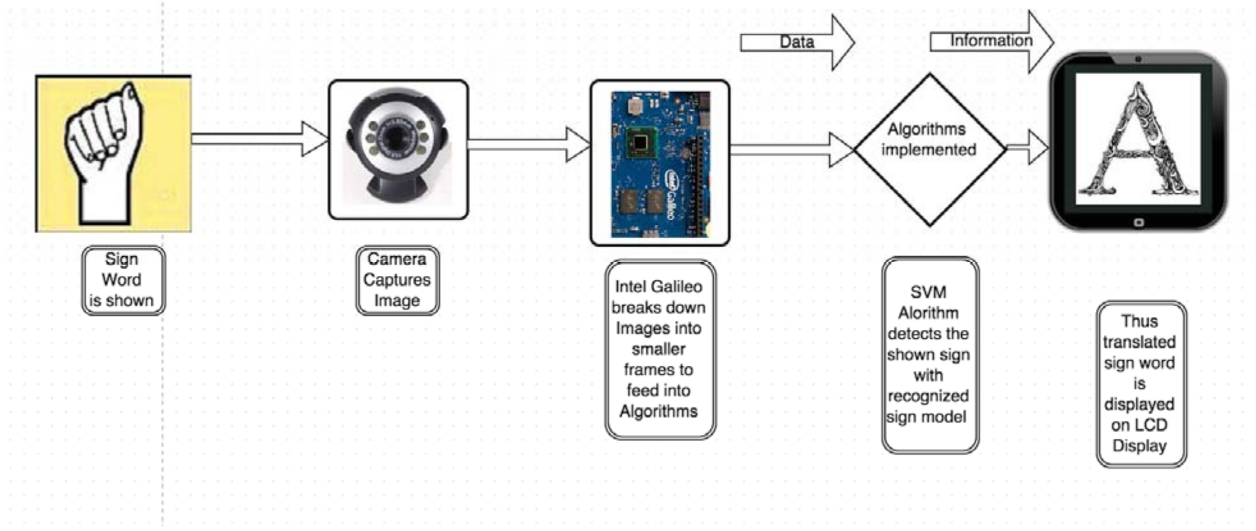
c. Processor Features

- 400 MHz 32-bit Intel® Pentium® instruction set architecture (ISA)-compatible processor
- 16 KByte L1 cache
- 512 KBytes of on-die embedded SRAM
- Simple to program: Single thread, single core,constant speed
- ACPI compatible CPU sleep states supported
- Integrated Real Time Clock (RTC), with optional 3V “coin cell” battery for operation between turn on cycles

d. Storage Options

- 8 MByte Legacy SPI Flash to store firmware (bootloader) and the latest sketch Between 256 KByte and 512 KByte dedicated for sketch storage
- 512 KByte embedded SRAM
- 256 MByte DRAM
- Optional micro SD card offers up to 64 GByte of storage
- USB storage works with any USB 2.0 compatible drive
- 11 KByte EEPROM programmed via the EEPROM library

2. Flow Chart


Data
Information
Sign Word is shown
Camera Captures Image
Intel Galileo breaks down Images into smaller frames to feed into Algorithms
Algorithms implemented
SVM Alorithm detects the shown sign with recognized sign model
Thus translated sign word is displayed on LCD Display

3. System Architecture/ Software

Our first goal is to be able to interpret sign language alphabets which are one image without motion. The translation of these signs can be done with one image processing while the rest of the sign language needing the motion of the hand to be processed also. The second part/ goal of the project is to be able to train our device to these sign with motion and be able to interpret them.

When recognizing signs of the alphabet that do not require any motion but just an image, we can use the device camera to take a picture instead of a video and recognize the sign. We normalize the image of the hand by steps that include centering the hand from right to left, getting rid of the background, changing the image to binary (black background and white hand), finding the smallest width of the hand from bottom up and setting that (the wrist) as the new bottom of the image, and getting rid of any noise that can affect the image processing to determine the sign in the image.

The design is based on machine learning mechanisms (SVM) to train our device on learning the sign languages and for it to be able to detect signs and interpret them correctly. The system will use multiple SVM systems for better accuracy. The first use of SVM is to recognize a hand gesture using Knight' algorithm of pyramid of histogram of oriented gradients. By using this as a feature for a SVM with 70-30 cross validation, we will be able to distinguish between a hand and a non-hand. This helps us determine the hand from the background and makes it easier to process the image. Using Sharif's algorithm for identifying hand configurations with low resolution depth sensor data, we can apply a combination of 2D shape based and size based features to recognize the configuration of the hand in the scene. Tracking the data gathered by the camera and using Marx's SVM based system, we can clarify between one finger, two fingers, a closed fist, and open palm gestures. By collecting the relative area, height and width of the normalized hand image we can determine the width at the top of the hand and the gap at the top of the hand. Using these and maybe additional systems for SVM we can train our device to be able to normalize an image input and recognize it so it can be translated and outputted by voice or text. If there is an error in the initial sign put into it, it will also be able to recognize that and show the user the proper method of performing the sign.

When the sign is a motion and not just one image, we will use systems of machine learning to perform the needed steps to recognize all images during the motion .The motion of the hands will can be detected in real time by comparing the grey scale intensity of two consecutive image frames. The centroid of the difference picture at frame will represent a good approximation on hand position. It will be done by tracking and storing hands' position, and keeping only the image area around the hand at the stop locations. Then, we can train our device to recognize these motions and, along the image processing part of the system, be able to recognize the sign and translate it, recognizing most of the 6 thousand signs.

4. Final Schematic Description

The initial design proposed is to build a device (1) to translate and interpret ASL into English language by using a camera (3) to realize the signs. One conceptual design consists to use a camera that will identify the attempted sign language and if the sign is inaccurately executed, the display (2) will correct each sign for the user from pictures already stored in the database. Each signal will have around 10 similar pictures that will be used like model into the database. Then this pictures are used to compare with the future signals of the user (inputs) to give the desired and correct answer (outputs).

To start the process, it is necessary to press the button ON/OFF (4) that turn on the device. The camera (3) will take pictures immediately after some movement, then analyze each picture with the signs stored in the memory. If the first picture did not consider similar (70% of accuracy), the second picture is immediately analyzed (keeping the operation fast). To input an accurate and readable sign language, the Zoom Buttons (6) can be used to manipulate the focus and size of the images. The Zoom Buttons (4) allow the user to zoom in up to 3 different levels and also zoom out up to 3 levels.

If some sign input is misunderstood, the user can input it as a text into the device. This method, using the keyboard (1), will take in a text message as an input and output a voice and image output. The voice output of the message comes out through the speakers (8) and the image/video output is displayed through the LCD Screen (2). To keep the quality of the speakers (8), the user can use the Volume buttons (5) to increase or decrease the volume of the sound. This device will be charged via USB port (7). This will also serve as our communications port where we will be able to add/remove data to the systems storage.