Sign Language Emulating Robotic Assistant (*SLERA*)

Made by:

JD Ritchey

Tim Campbell

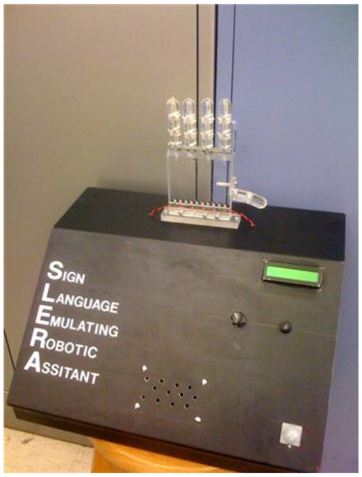
Nick Haub

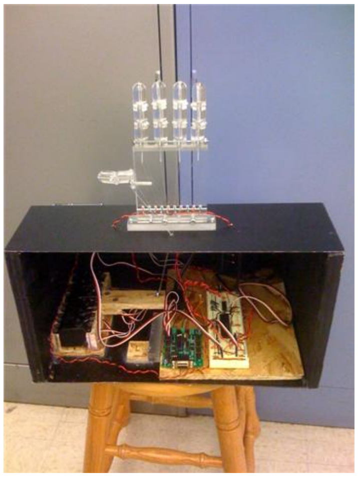
Brian Taylor

Design Summary

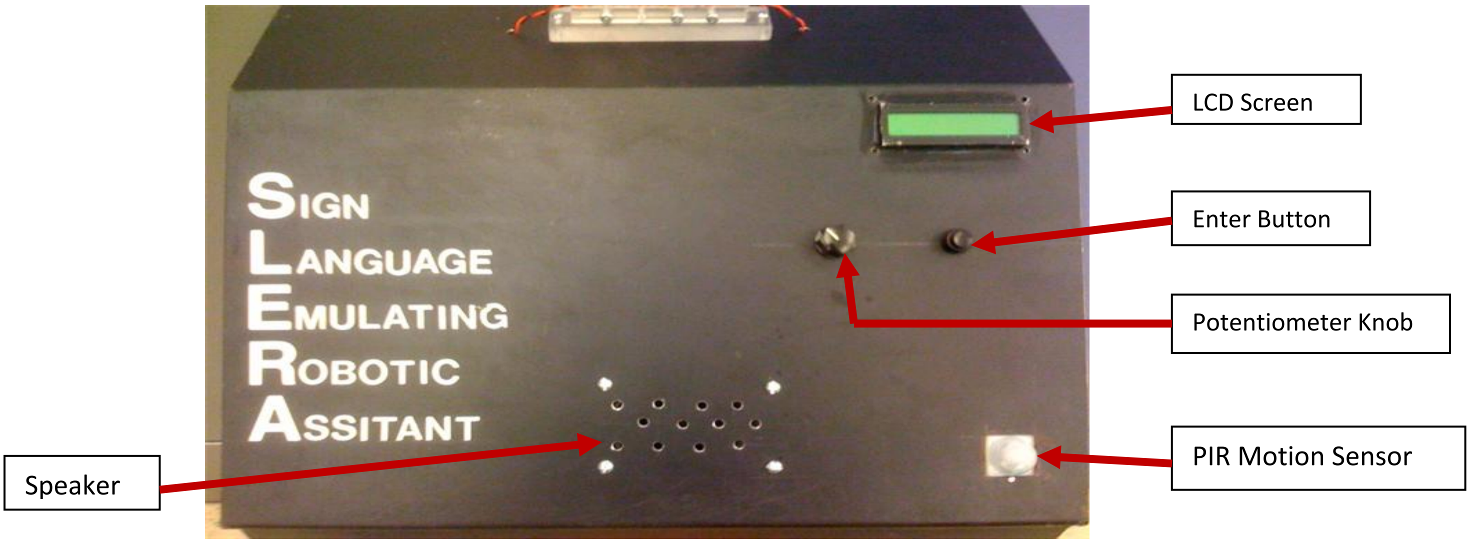
(See figures F1, F2 and F3 for italicized components)

This project consists of an acrylic hand that can display sign language. The hand has all the digits, or 'bones' of a human hand, and while not having the same range of motion, is able to simulate sign language for teaching purposes. To initialize the hand, the *PIR motion detector* detects when someone approaches and starts a welcome sequence by waving the hand, turning on the white LED's and giving directions for use on the *LCD screen*. Input comes from the *potentiometer knob* and the *enter button*. Moving the potentiometer knob moves the arrow beneath the different letters and symbols on the LCD screen. Once a letter or symbol is selected and the enter button is pressed, six *servo motors* pull the '*fingers*' to the proper positions. When digits are in their proper places the *speaker* says the letter via the SpeakJet pre-manufactured IC. The *PIC16F917 Microcontroller* logically interprets information from the PIR motion sensor, potentiometer knob and enter button. The PIC then serially communicates with a *Lynxmotion SSC-32 Servo Controller* which, in turn, communicates with each of the six servo motors individually. The fingers are pulled by standard servo motors. *Ten lb.-test fishing line* is connected on each side of the finger from the finger-tip to the servo motor. When the fishing line is pulled, the finger will curl or straighten depending on the servo motor direction. There are four motors to control the curling and straightening motion of the fingers and two motors to control the dynamic motion of the thumb. (The figures below show the overall layout.)

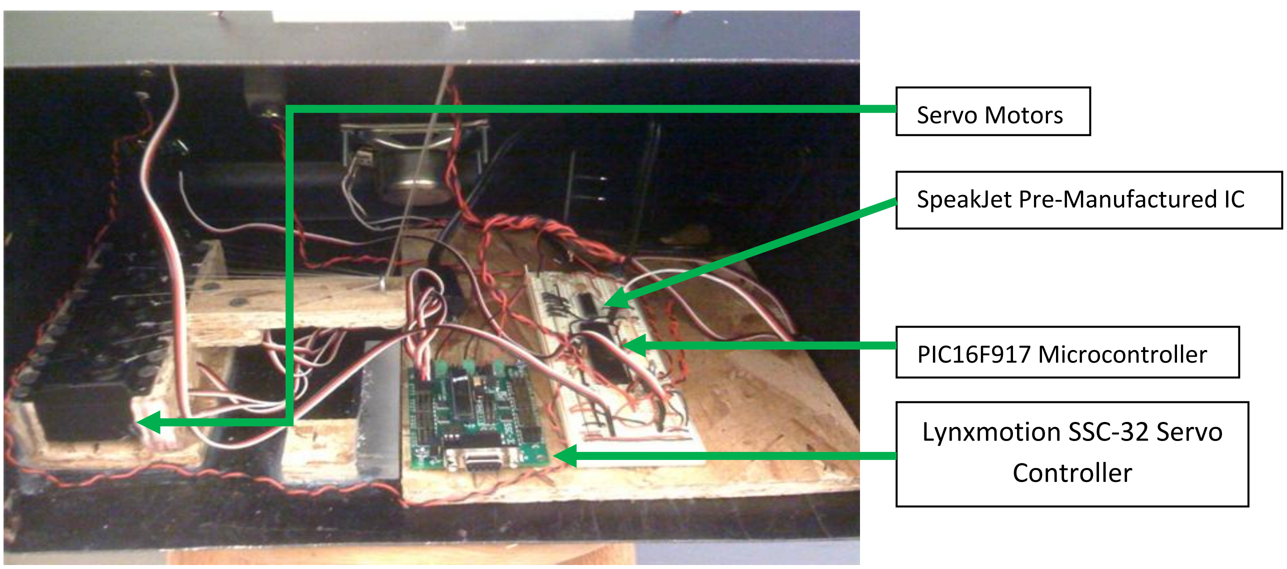




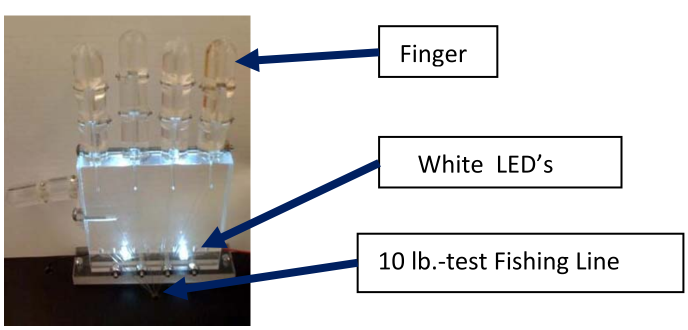
F1

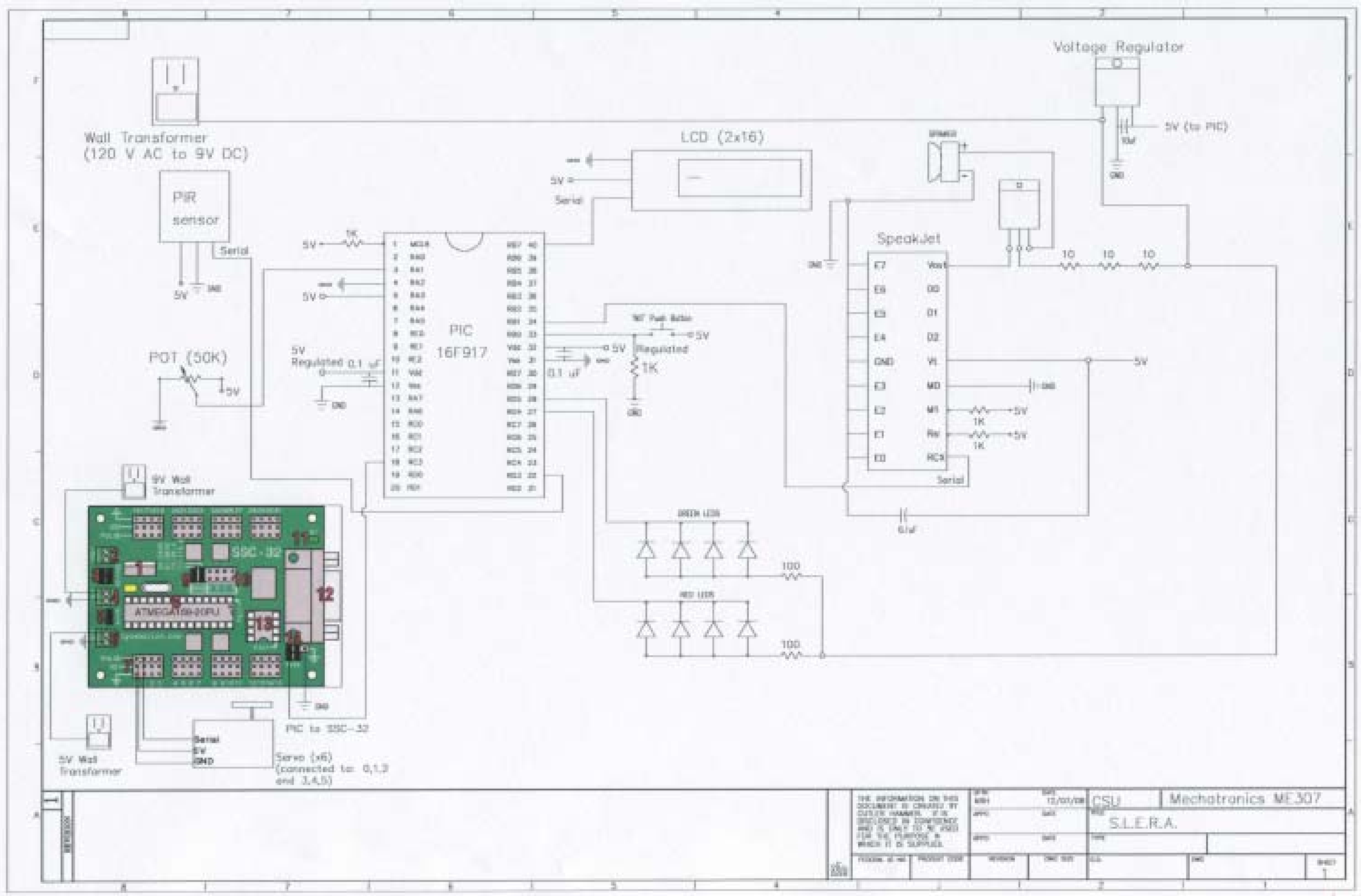

LCD Screen
Sign
Language
Emulating
Robotic
Assitant
Enter Button
Potentiometer Knob
Speaker
PIR Motion Sensor

F2

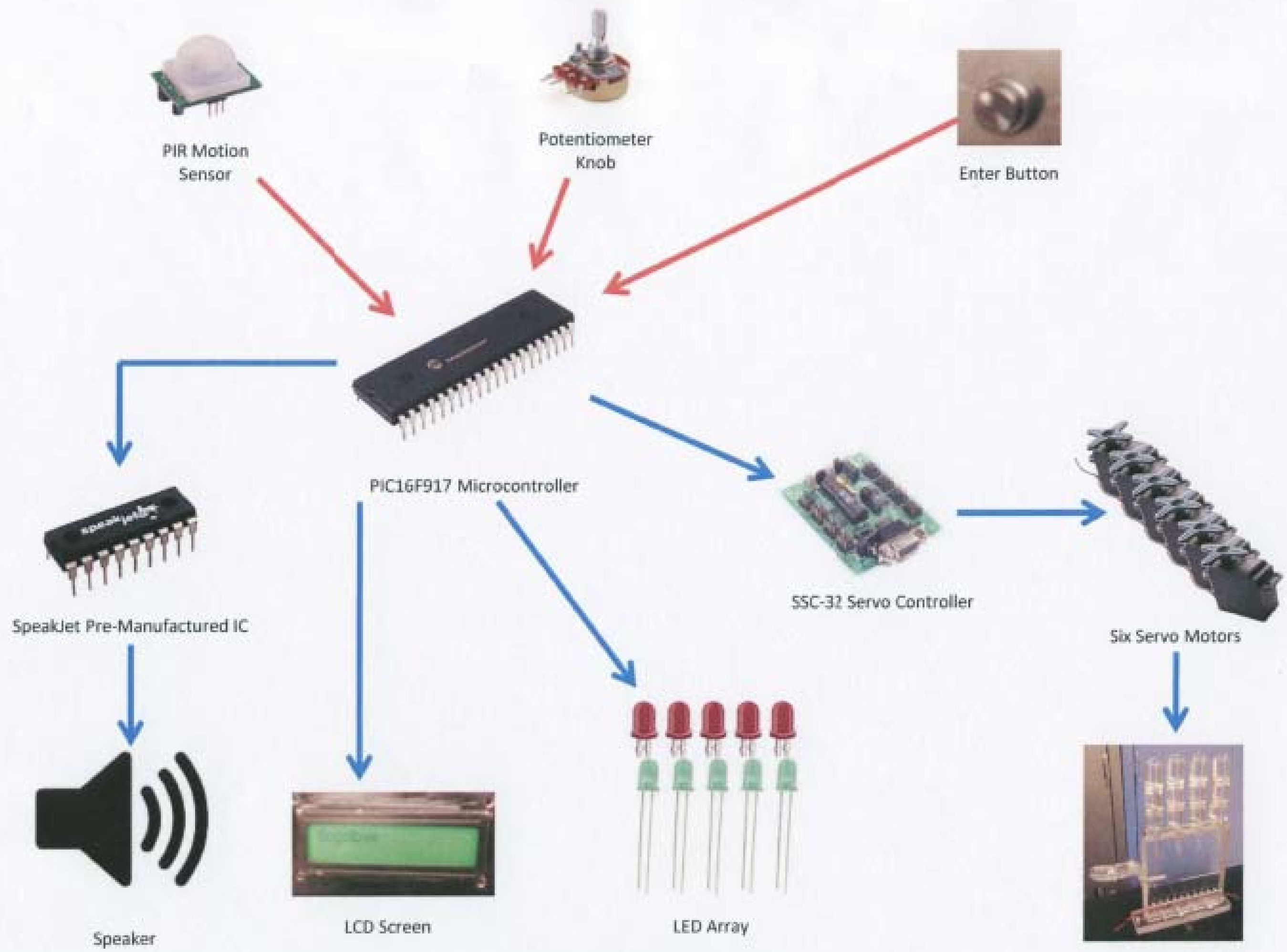

Servo Motors
SpeakJet Pre-Manufactured IC
PIC16F917 Microcontroller
Lynxmotion SSC-32 Servo Controller

F3


Finger
White LED's
10 lb.-test Fishing Line



Voltage Regulator
5V (to PIC)
GND
Wall Transformer
(120 V AC to 9V DC)
LCD (2x16)
5V
Serial
PIR
sensor
Serial
SpeakJet
PIC
16F917
POT (50K)
5V
Regulated
0.1 uF
1K
10
100
5V
GREEN LEDS
RED LEDS
9V Wall
Transformer
5V Wall
Transformer
SSC-32
ATMEGA168-20PU
PIC to SSC-32
Serial
5V
GND
Servo (x6)
(connected to: 0,1,2
and 3,4,5)
CSU
Mechatronics ME307
S.L.E.R.A.



Functional Diagram

Design Evaluation

Output Display:

Our output display is a backlit 2 x 16 character LCD. Initially we assumed we would need at least 8 output pins from our microcontroller to run our display. We later found that only one was needed to send data serially. Therefore our LCD turned into a relatively simple component to integrate. We also incorporated two ultra bright white LED's. These stay on during operation to light up the hand. The light reflects and diffracts off all the different angular planes and surfaces to visually stimulate our user. We created some custom characters for some special hand gestures. This involved sending nine bytes of data at the beginning of the program to be saved in the LCD's RAM. These nine bytes tell the LCD which pixels to send high and thus create whatever display we desire.

Audio Output Device:

We took apart an old computer speaker for our audio output. We run this from a pre-manufactured IC from the SpeakJet Corporation. This IC is connected serially to our main brain PIC. After extensive research we found that with combinations of serial outputs we can essentially output nearly unlimited phrases and sound effects. We ran into complications with the volume of the speaker, but with careful resistance and voltage manipulation we found a suitable volume. We configured the SpeakJet to say all of our projects audio needs, including developing a sound pattern to create our name.

Manual Data Input:

We initially planned to use a full keyboard for our manual data input. This was a humbling experience. After much research, we decided to review our design. We found it cheaper, simpler, and more practical to use a potentiometer with our PIC's internal A/D converter.

Automatic Sensor Input:

Automatic sensor input was achieved with a PIR infrared proximity sensor. This functions as our primary initiation. When an infrared being (e.g. a person or animal.) comes within the sensors range our hand acknowledges and acts accordingly. It also conserves energy by acting as a sensor for the shut down timer. Three pins connect to ground, five volts, and data. Our main brain PIC receives the digital data without any need for A/D conversions or amplifications.

Actuators:

Actuation is controlled by six servo motors, five of which control the inward and outward bending of the fingers. The last servo controls a second degree of freedom for the thumb. We had a substantial amount of iterations throughout the design process. The kinematics of our hand proved to be complex enough to warrant reduction of degrees of freedom. We paired the motors down from nine to six. The individual joints of our fingers were also quite complex to design. They required a lot of machining, but the final product proved to be worth the trouble.

Logic, Counting, Integration, and Control:

The logic design for this project was mainly determined by the other categories and constraints. Programming followed those other constraints and, therefore, did not require time for specific logic design. So in this situation, when a problem arose in the programming, it was immediately dealt with and did not require multiple iterations in the function of the program.

There are four main loops in our program that all intertwine. Along with these main four, there are five subroutines that are called out in the main loops. This complicated the software flowchart. At startup, after defining variables and registers, a sixty second PIR calibration occurs. When the calibration is complete (marked by a display on the LCD), a small loop polls for a “high” from the PIR. When the PIR goes high, the main startup loop will execute. During the PIR polling loop, interrupts are enabled such that a press of the NO button will also send the program to the startup loop. In the startup loop, our main brain PIC sends serial commands to the SSC-32 Servo Controller to make the hand wave. From there, the LCD lights up and displays our welcome sequence including the name of our project and our names. Interrupts are disabled in this section of the program to prohibit improper function.

The functionality of this device comes from the main menu. It consists of letters and special characters on the top row of our LCD and an arrow on the bottom row that moves to any letter. The arrow moves with the turn of a potentiometer that is connected through an internal A/D converter. If a voltage value is within a range that we calculated for each letter, the arrow will move to directly below that letter. To have an active display we had to refresh our screen for every A/D conversion cycle (approximately every 30 milliseconds). Because the amount of gestures exceeded the number of cells on the top row of the LCD, a second display screen was needed. The first screen, our row of letters, has a blank cell followed by a right-facing arrow on the far right. When the selection arrow is positioned in the cell directly below the right-facing arrow, the screen automatically updates to show the second screen of alternative gestures. This process is reversed to return from the second screen. Interrupts are enabled again, but this time the interrupt NO button executes the actuation routine. The actuation loop utilizes the screen configuration as well as the cell number of the selection arrow to produce the calculated motion. If…Else statements give the servos predefined position values which are then sent serially to the servo controller. After the servos complete their motion, serial data based on the above is sent to the SpeakJet for our audio output. After the SpeakJet has spoken the prescribed letter, the program returns to the main menu.

From the main menu loop, if the selection arrow does not change position and if the PIR does not sense motion, a shutdown timer will be incremented. After twenty seconds of no motion, our shutdown sequence begins, which says “goodbye,” displays “Shutting down…,” and closes the hand. The LEDs stay on to provide visibility and reduce the risk of accidental damage due to its transparent nature. The program is then looped back to the PIR sensing loop poll for motion again.

Partial Parts List

PIC16F917

Description: This is an 8 bit, 40 pin DIP with an internal A/D converter and 256 bytes of EEPROM.

Model Number: PIC16F917

Source: Microchip

Allied Stock#: 383-1239

Price: $3.39

SSC-32 Servo Controller

Description: This controller allows the PIC16F917 to precisely move up to 32 servo motors simultaneously according to speed or time.

Model Number: SSC-32

Source: Lynxmotion

Price: $39.95

SpeakJet

Description: This is a voice and complex sound synthesizer self-contained within a DIP. It allows users to string together tones to produce words or sounds.

Model Number: R225-SPEAKJET

Source: Acroname Robotics

Price: $24.99

PIR Infrared Motion Sensor

Description: This is an infrared motion sensor with a 20' radius that uses a simple three-pin connection and a single bit output.

Model Number: 555-28027

Source: Parallax

Price: $9.99