CHAPTER IV

Software Design

4.1 Theory of System Operation

Figure 3.2 in previous chapter shows the system block diagram showing the interconnections between each block or module. All the modules are mounted onboard as to ease the prosthetic arm movement. This includes a voice recognition module which is located nearest to the user so as to make it handy and easy to use.

For the power source, it uses a 3.3, 5 and 12 V DC which comes from a power supply module.

Generally, the input voice level and ambient noise affects the recognition accuracy result.

For best recognition result, the microphone should be mounted or attached as closed as possible to the user's mouth. Principally, the system is triggered by the voice command word produced by the user through the use of microphone. The user commands' for the arm movement by producing words which have been stored previously in the SRAM memory. This SRAM I. resides in the voice recognition processor. To keep the system as simple as possible, the words are kept short and the quantity is kept to minimum quantity. The quantity of words can be added and upgraded later on for future development and improvement. The eight basic command words are chosen and they are shown in Table 4.1.

The voice from the user is picked up by a microphone and the analog output of the receiver is then fed to the voice recognition module. In this module, the signal is then compared and matched to the data previously stored in its memory to determine the corresponding output command. Then it latches data which is in binary-coded decimal to input port D of the PIC microcontroller.

This BCD signal is then processed by the PIC and the output is sent to port B which is connected to the Servo motors. These signals will drive the motors and make the hand moves.

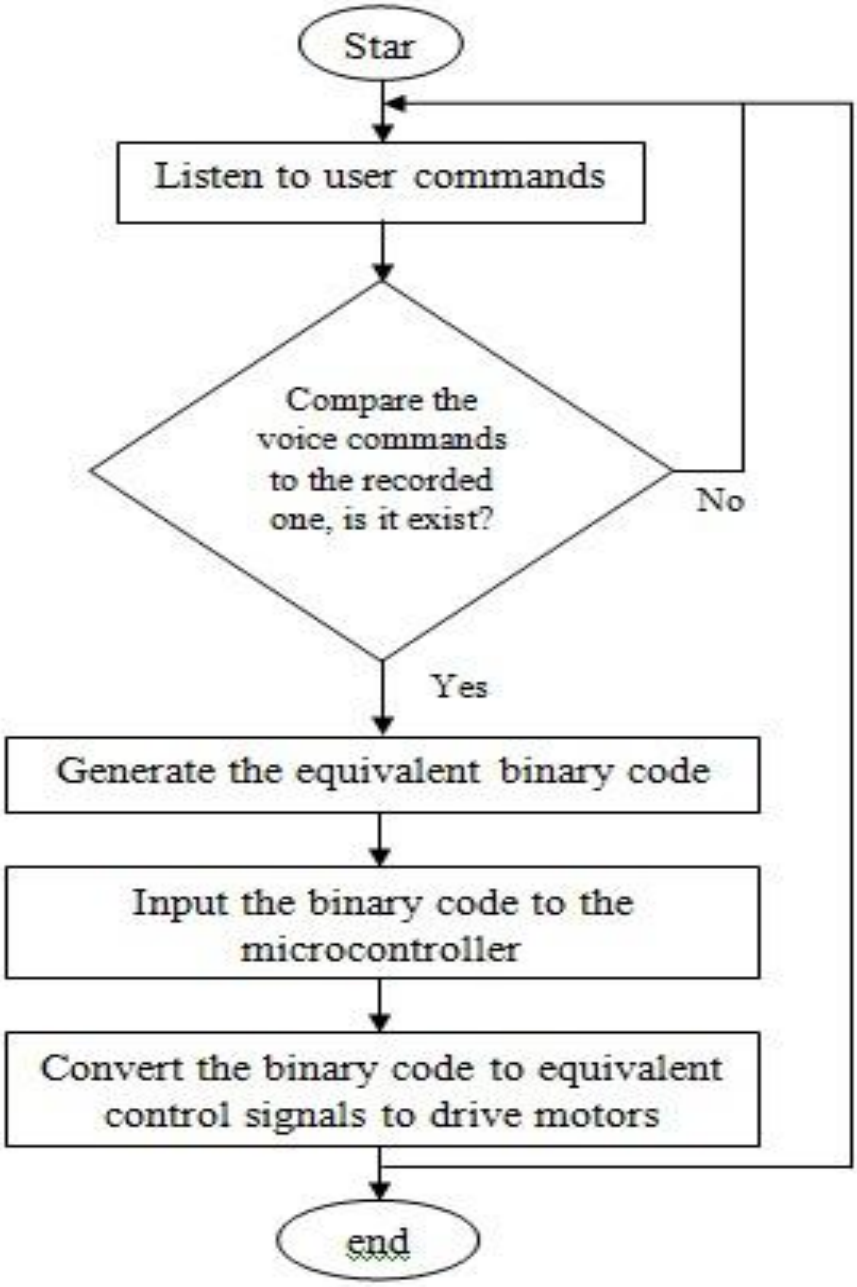

Figure 4.1: system operation flowchart

4.2 Software Design

The software development processes or task of the system have to undergo several steps and stages and it shown in Figure 4.2

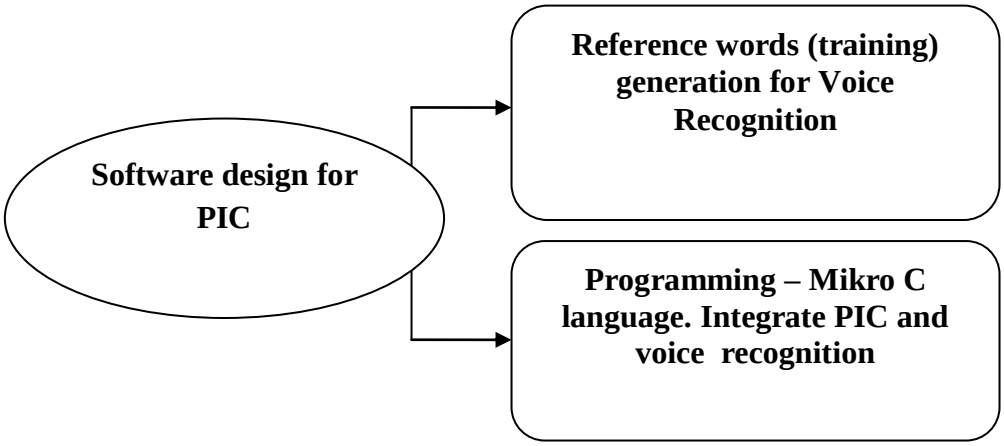

Figure 4.2: System software design steps

4.3 Voice Recognition Module Initialization

Once we need to train the SpeakUp to obey the commands. We must Plug in the board to the PC through USB cable. Configure it using the free software. Alternatively you can configure the board directly using the on-board buttons without using the software.

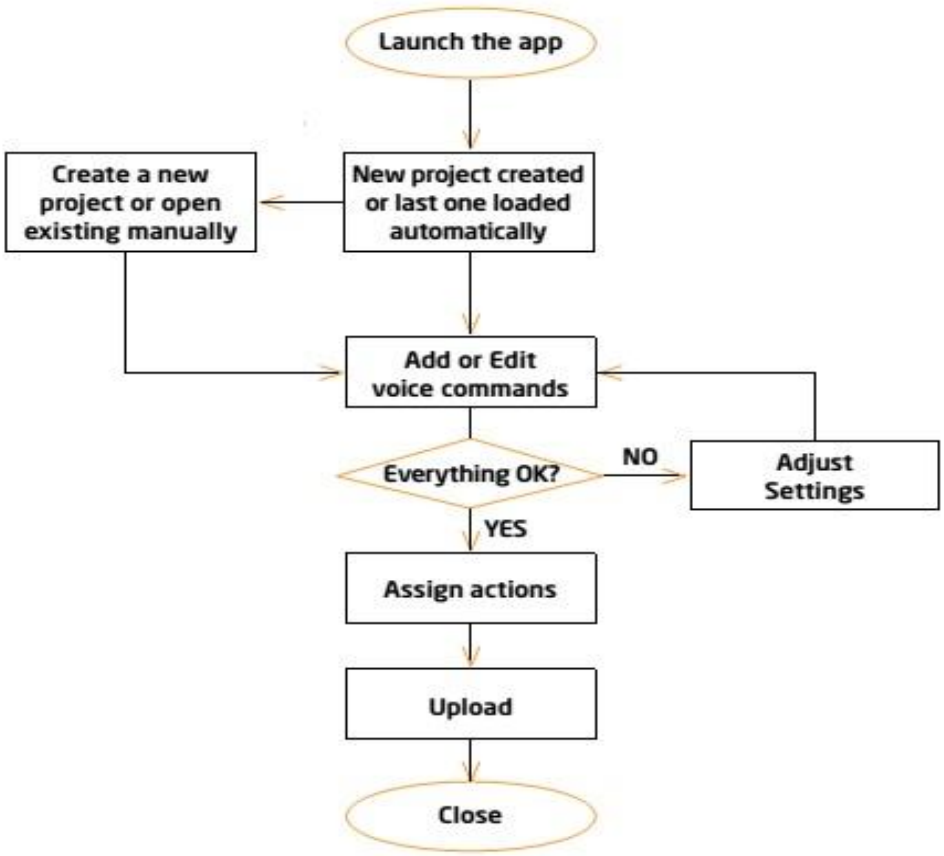

Figure 4.3: Voice Module initialization flow chart

Flow these steps to configure and use the voice module:

1) Connect the SpeakUp click board to the computer via the USB cable. It will be recognized as a USB Human Interface Device (HID) in the Device Manager of the Control Panel.

2) Once you connect the SpeakUp to your computer you're just a few clicks away from configuring it. The set-up process is dead simple. Launch the application, and it will lead you through the initial steps of recording and assigning commands.

3) To create a new project, press the Create New Project button from the main toolbar of the SpeakUp software.

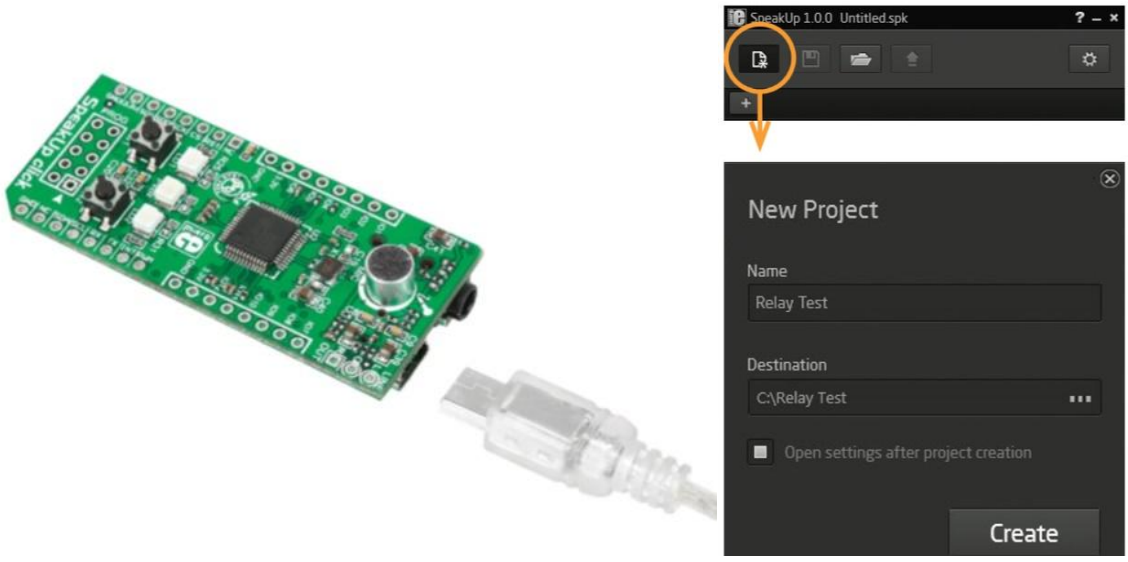

Figure 4.4: plug in USB cable and start a new project

4) A new window will open, where you can enter your project's name and destination folder.

5) To finish project creation after inputting the required information, press the Create button.

6) To record a new voice command, press the Add New Voice Command button.

7) A New Voice Command dialog window will appear. Press the Record button.

8) The recorded command will be played back automatically, so you can make sure it's OK.

9) If you're satisfied with the recording, enter a name for your command and click the Save & Close button.

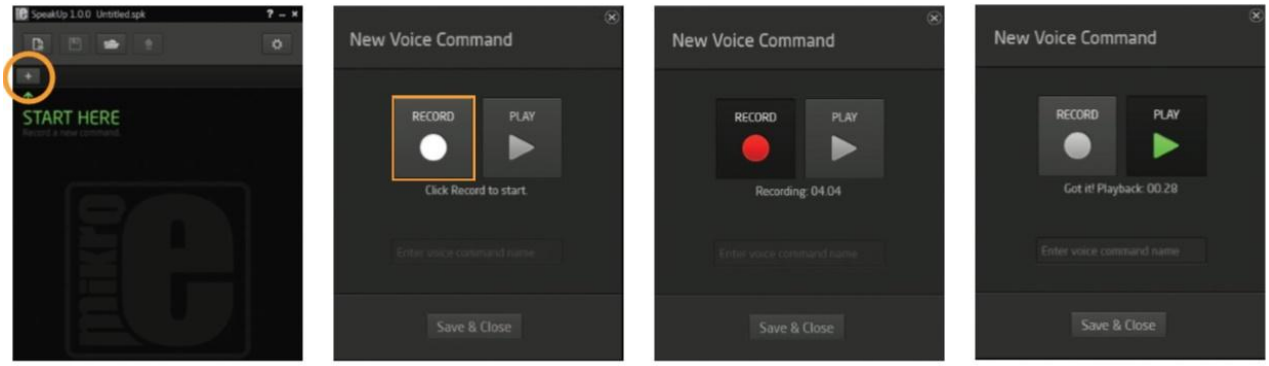

Figure 4.5: Record and save a new voice command

10) The recorded command will appear as a new tab. You can play it back, edit or delete it anytime.

11) Set pin aliases and Initial Pin States, you can rename GPIO pins according to your needs and set their starting conditions. The new GPIO pin aliases will be applied in the main window too. Set the corresponding initial GPIO pin state in the Initial Pin States section. Condition can be either low (logical 0) or high (logical 1).

Figure 4.6: Set the Initial Pin States

12) When a new command is recorded, it is time to assign it an action. The action will be performed when the voice command is recognized.

NONE: When this option is selected, no action will be performed on the corresponding GPIO pin upon voice command matching.

ON: When this option is selected, a corresponding GPIO pin will be set to logical high state upon voice command matching.

OFF: When this option is selected, a corresponding GPIO pin will be set to logical low state upon voice command matching.

TOGGLE: When this option is selected, a corresponding GPIO pin state will be toggled upon voice command matching.

PULSE: When this option is selected, a train of pulses will be sent to the corresponding GPIO pin upon voice command matching.

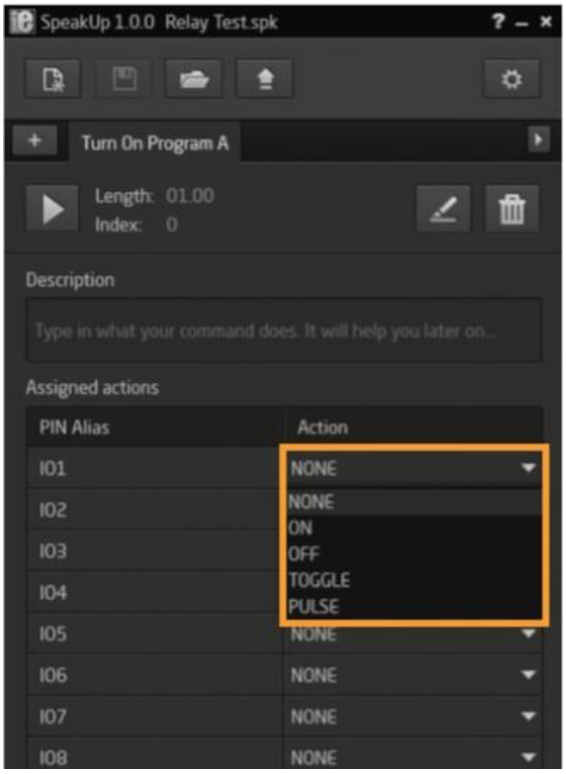

Figure 4.7: Assign pin actions

13) When you’re finished recording and configuring voice commands, it is time to upload the project to the SpeakUp click™ board. This is done via the Upload Project button.

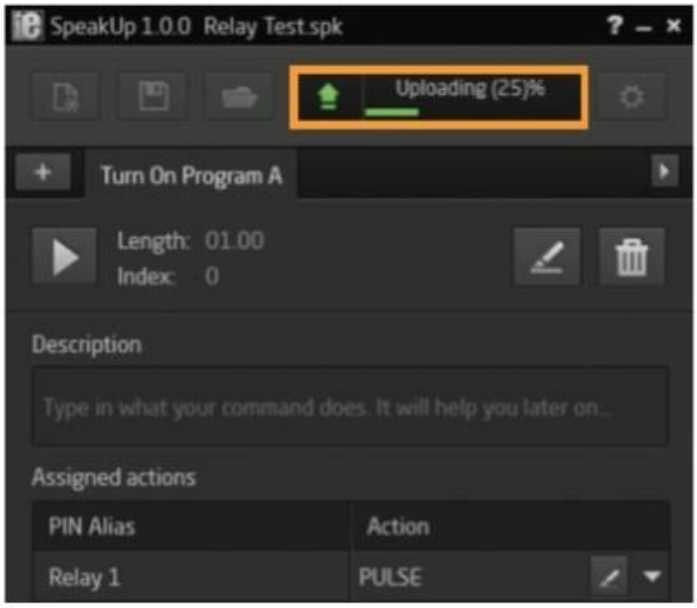

Figure 4.8: Upload the project code

Now the voice recognition module is ready to work and generate actions as response when voice command recognized.

4.4 Control System Software

In order to ensure that the written software works efficiently with the hardware, a system simulation is carried out in the Windows based environment called 'Proteus' (figure 4.9).

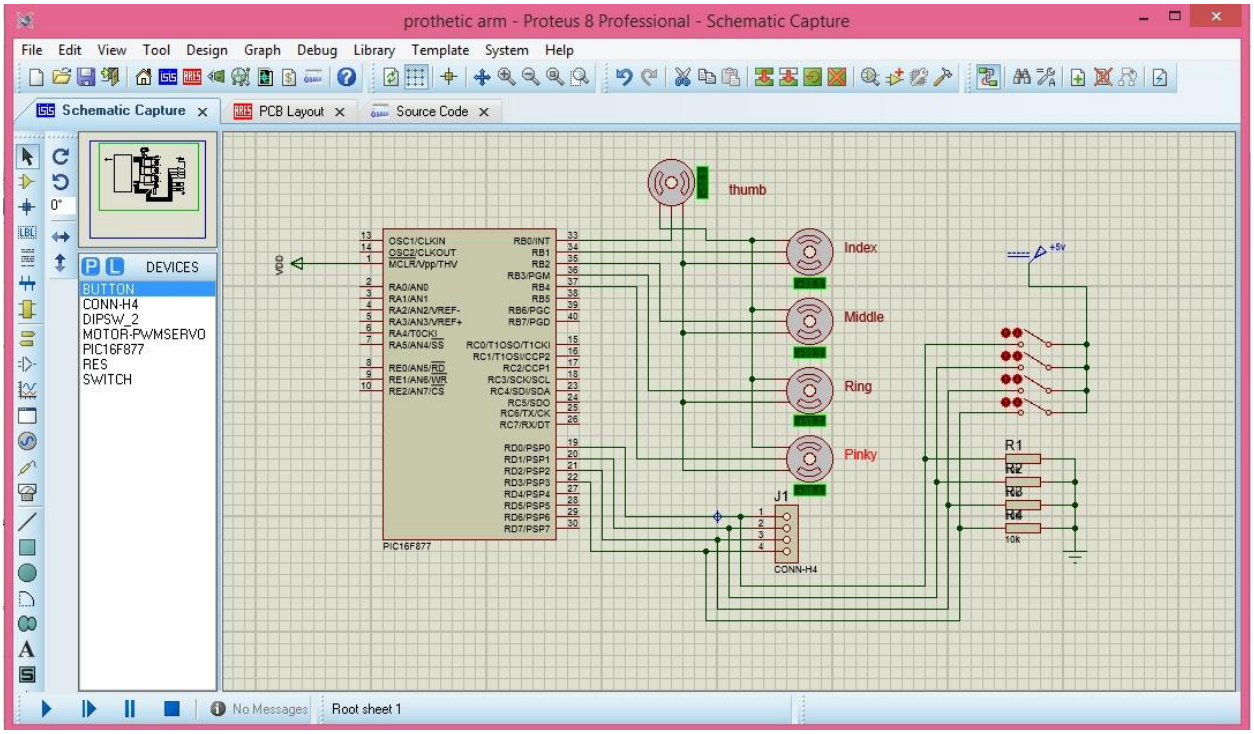

Figure 4.9: Project simulation on Proteus 8 environment

The software development process involves (1) witting and debugging an Mikro C language program as a main control program according to the PIC word instructions, (2) simulating it in Proteus environment (3) embedding it into the PIC microcontroller and (4) testing with the whole system. The software is written and developed in a Window-based environment called Mikro C Pro for PIC (figure 4.10) where it is converted to machine code for embedding process.

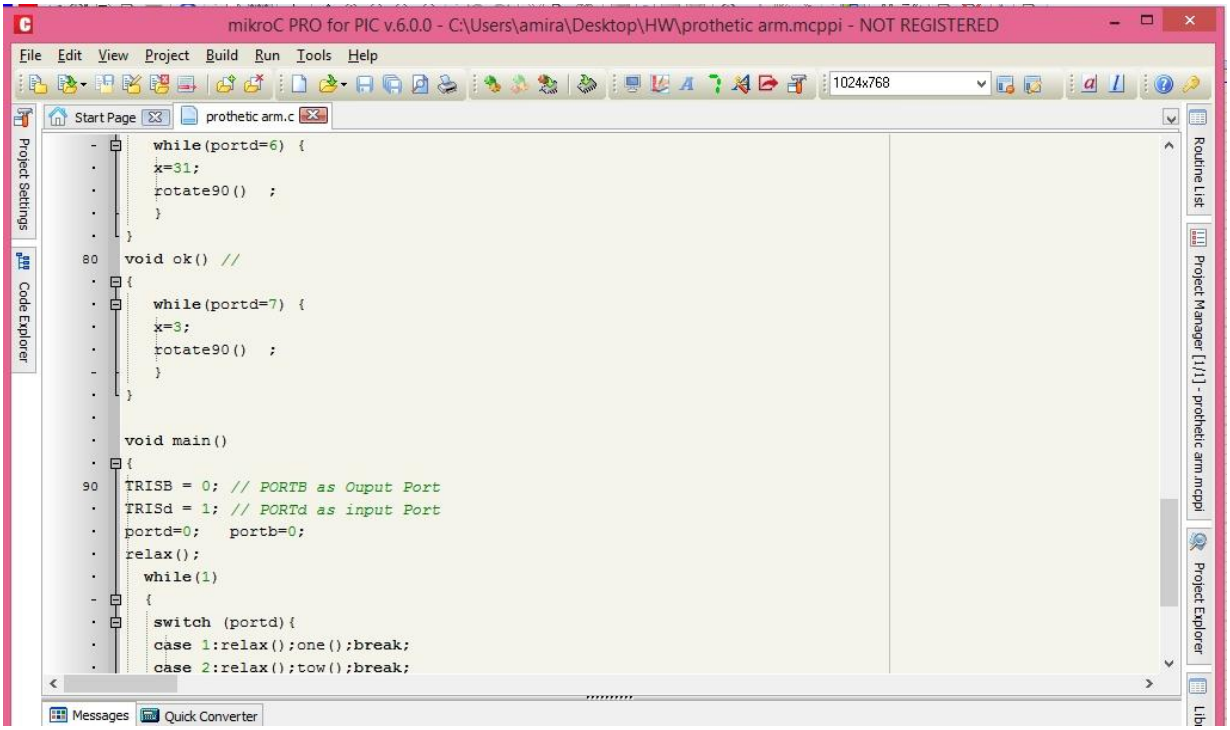

Figure 4.10: Writing MikroC code on MikroC pro

In the embedding process, a device called a 'programmer' is used to program or transfer the software into the PIC microcontroller. (figure 4.11).

Figure 4.11: Mikroprog programmer

After the above process completed, then the prosthetic arm control system is tested and evaluated. It is evaluated for the accuracy of the command-to-output result which is the voice recognition performance. A total of 8 set of data are to taken for this evaluation.