# CHAPTER 3

# METHODOLOGY

1. Firmware Design Process
2. Hardware Design Process
3. Software Design Process

In order to make the USV easy to operate, the use of a graphical user interface was of great assistance. The existence of controls such as menus, toolbars, buttons, and sliders in MATLAB GUI makes it convenient for a wide range of applications. In this project, the MATLAB GUI was used to implement the software application for monitoring and control.

1. Graphical User Interface

The graphical user interface design mainly consisted of user control panel, video frame and monitoring panel. It had additional features: namely battery status and time. Special buttons were also provided to allow the user to hide/show the video frame or capture images from the video frame. The captured images were automatically saved into a designated file.

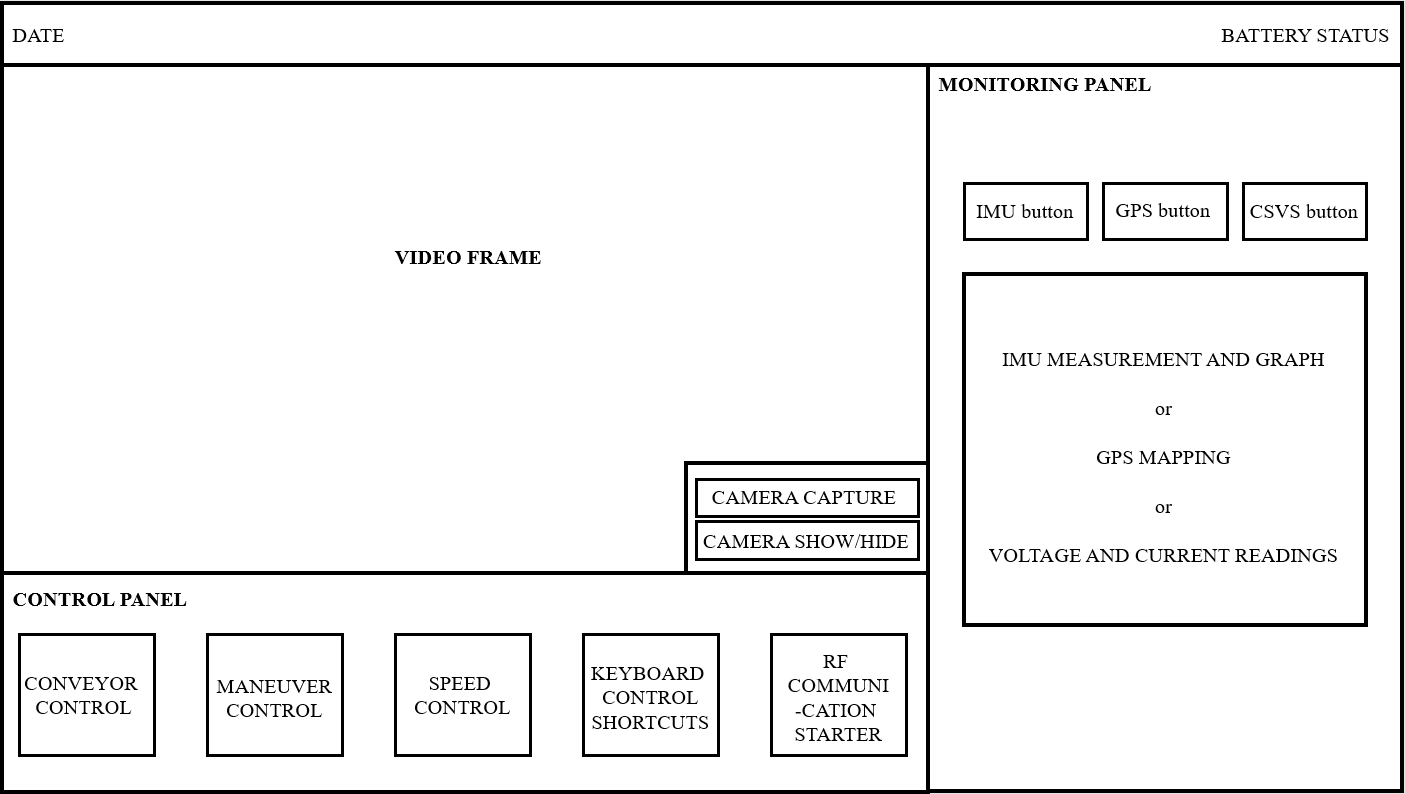


Figure 3.3.1 Human-Machine Interface

1. Serial Port Communication

To receive and transmit data, the software application utilized the ‘serial’ function to create a serial port object with the specified property name and property value. The baud rate was set to 9600, similar to the USV. The COM port is selected by the use of the ‘instrhwinfo('serial')’ function which provided the available serial hardware. To start obtaining and editing data, the function ‘fopen’ is used.

Once the communication between the USV and graphical user interface was established, receiving and transmitting data were done using the “fscanf” and “fprintf” functions, respectively. The ‘fscanf’ function reads the data from the open text file. The ‘fprintf’ function returns the number that ‘fprintf’ writes to the open text file, using any of the input arguments in the preceding syntaxes. Additionally, in order to sort the string of data received by the software, the ‘strtok’ function was used. This function parses string from left to right, using the specified character, in this case, ‘,’ as delimiters, and returns part of the text in token. Lastly, the data received was monitored and set to zeros (0) when it reached the maximum allowed rows and columns to avoid cluster of data.

1. Video Streaming and capture

In accessing the wireless camera from the USV, the ‘imaqtool’ command was utilized. The command showed all the available video input devices attached to the computer and its properties. This allowed the connection of the right video input. The video was then shown in an axis in the user interface.

A push button and a radio button were provided on the bottom-right corner of the video frame. The push button served to capture images from the video frame by the use of ‘getframe’ function. The images are then saved in a specified file in the computer with the name of the time format ‘yyyy-mm-ddTHHMMSSZ’. It was necessary to name the images this way, since every name should be unique for each image to avoid overwriting. While the radio button was provided to make the video frame appear or disappear.

1. User Controls

The graphical user interface consisted of buttons that allowed the user to control the USV. The set of buttons were either push buttons, toggle buttons or slider, depending on which is more convenient. Each of these buttons had a specific function and pressing it transmitted a specific character to the USV through wireless serial communication to perform its corresponding function. In addition, shortcut keys to these buttons were provided to enhance accessibility, shown in table 3.3.1.

3.3.4.1 Maneuver Control

For maneuvering, push buttons allowed the forward, stop, left, right and center movement of the USV. Pressing these buttons sent data to the firmware to perform the specific maneuver command. Sending a ‘W’ moved the USV forward, ‘A’ turned it to the left, ‘D’ turned it to the right, ‘Q’ returns it to center, and ‘S’ stopped it.

3.3.4.2 Conveyor Control

A toggle button was used to control the conveyor of the USV, since it was either on or off. Pressing this toggle button sent data to the firmware to perform the corresponding command. Sending a ‘C’ turned the conveyor on, while sending a ‘V turned the conveyor off.

3.3.4.3 Propeller Speed Control

A slider is used to control the propeller’s speed. The length of the slider is divided into 9 corresponding speed. From bottom to top of the slider, the corresponding speed is 0 to 9, respectively.

3.3.4.3 Keyboard Control Shortcuts

By the use of ‘keypressfcn’ function of MATLAB, use of shortcut keys were made possible. This made it more convenient to control the USV. Edit text was provided in the control panel to input the specific letters and numbers assigned for control.

|  |  |  |
| --- | --- | --- |
| Data Sent | Shortcut Keys | Action |
| W | w | Move USV forward |
| S | s | Stop USV |
| A | a | Turn USV to left |
| D | d | Turn USV to right |
| Q | q | Turn USV to center |
| C | c | Turn conveyor on |
| V | v | Turn conveyor off |
| 0 | 0 | Set propeller speed to 0 |
| 1 | 1 | Set propeller speed to 1 |
| 2 | 2 | Set propeller speed to 2 |
| 3 | 3 | Set propeller speed to 3 |
| 4 | 4 | Set propeller speed to 4 |
| 5 | 5 | Set propeller speed to 5 |
| 6 | 6 | Set propeller speed to 6 |
| 7 | 7 | Set propeller speed to 7 |
| 8 | 8 | Set propeller speed to 8 |
| 9 | 9 | Set propeller speed to 9 |

Table 3.3.1 Control Commands

1. Sensor Data Interpretation

Through wireless communication, strings of data were sent from the USV to the ground station by the firmware. These strings of data were decoded by the software. As mentioned in section 3.3.2, these strings were parsed and then tokenized. The tokenized data was then represented through textual or graphical representation.

3.3.5.1 GPS Mapping

-Work in progress-

3.3.5.2 Battery Monitoring

-Work in progress-

3.3.5.3 IMU

-Work in progress-

In order to correctly evaluate the performance of the USV system, the sensors and control units underwent evaluation tests. In these tests, at least three (3) attributes was measured: namely, sensitivity, accuracy, and precision.

For the sensors, its sensitivity or the ability of the sensors to transmit changes in measurements in the quickest amount of time was tested. The ability of the sensors to give a true measurement, its accuracy, and its ability to give consistent results, its precision, are good measures to ensure valid data acquisition. The data provided by the sensors was compared with actual data measured from reliable devices. Then, the mean percentage of errors was calculated.

For the control units, its responsiveness to the command of the user, the ability to perform according to the task and give consistent results was tested.