Technical Manual

Principal of Operation

The Heads-Up Display is a device that serves to provide the user with information about the surrounding world. The information is displayed on a Nokia 3310 LCD display, which provides 84 pixel by 48 pixel resolution. The specific information displayed comes from a Venus 634LPx GPS unit, a 9 DoF (Degree of Freedom) Razor IMU (Inertial Measurement Unit), and a XL-Maxsonar EV0 Ultrasonic Rangefinder.

The data pulled from the GPS is used to display:

- Time (hours and minutes, 24-hour format)
- Current GPS position (latitude and longitude, to the nearest hundredth of a minute)
- Distance to the target position (feet to the north or south and feet to the east or west)

The data pulled from the IMU is used to display:

- Compass heading (cardinal and ordinal directions: N, NE, E, SE, S, SW, W, NW)

The data pulled from the sonar sensor is used to display:

- Distance to nearest object (meters, to the nearest centimeter)

The system is coded in the C programming language using the µCOS-II real time operating system. A display library is used to convert ASCII characters into 5 pixel by 7 pixel characters that are displayed at set locations on the LCD screen.

Design Diagrams

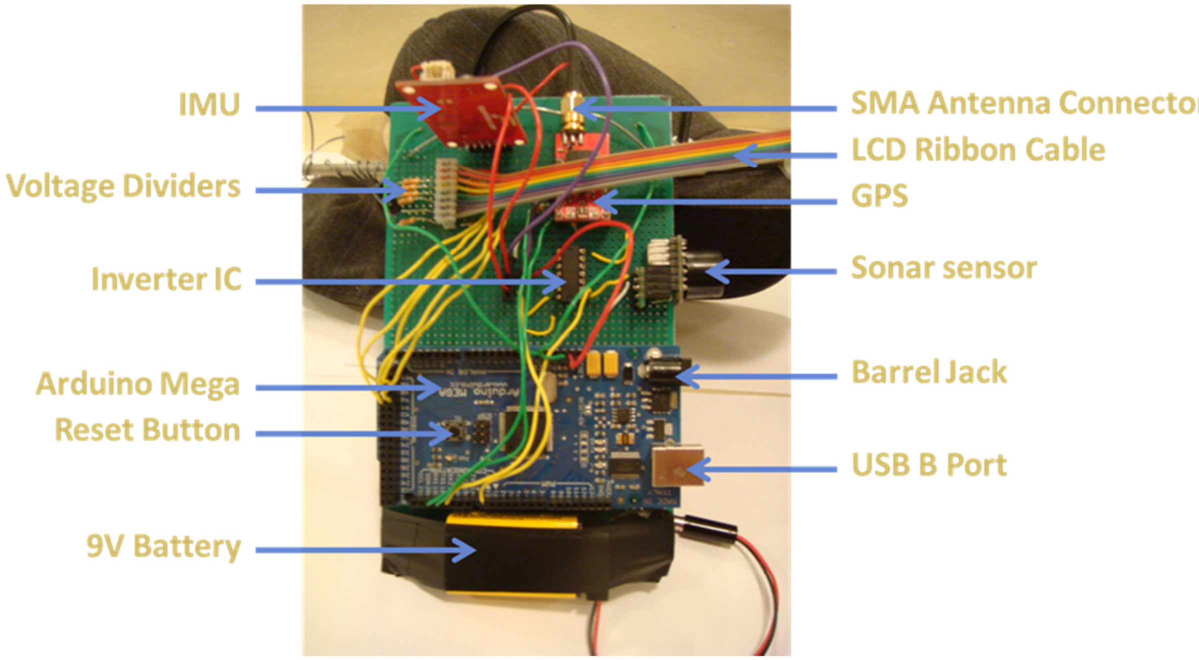

Figure 1: Side view of the Heads-Up Display's electronics board.

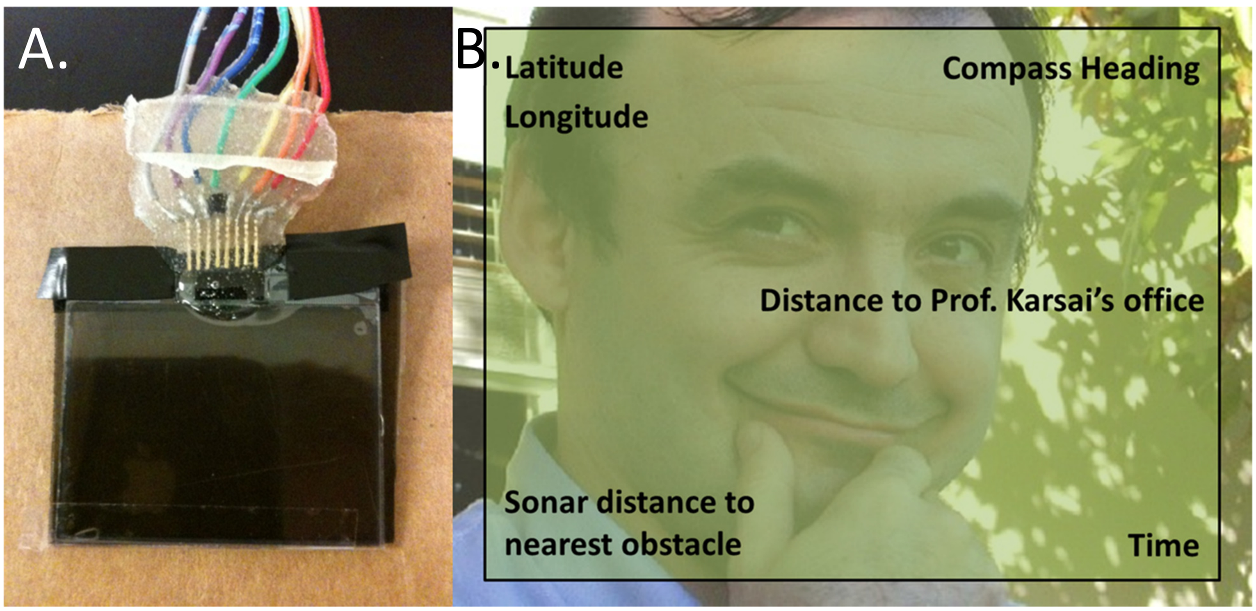

Figure 2: A) Nokia 3310 LCD Display; B) Layout of data display on HUD screen

Explanation of Design

Structural Design

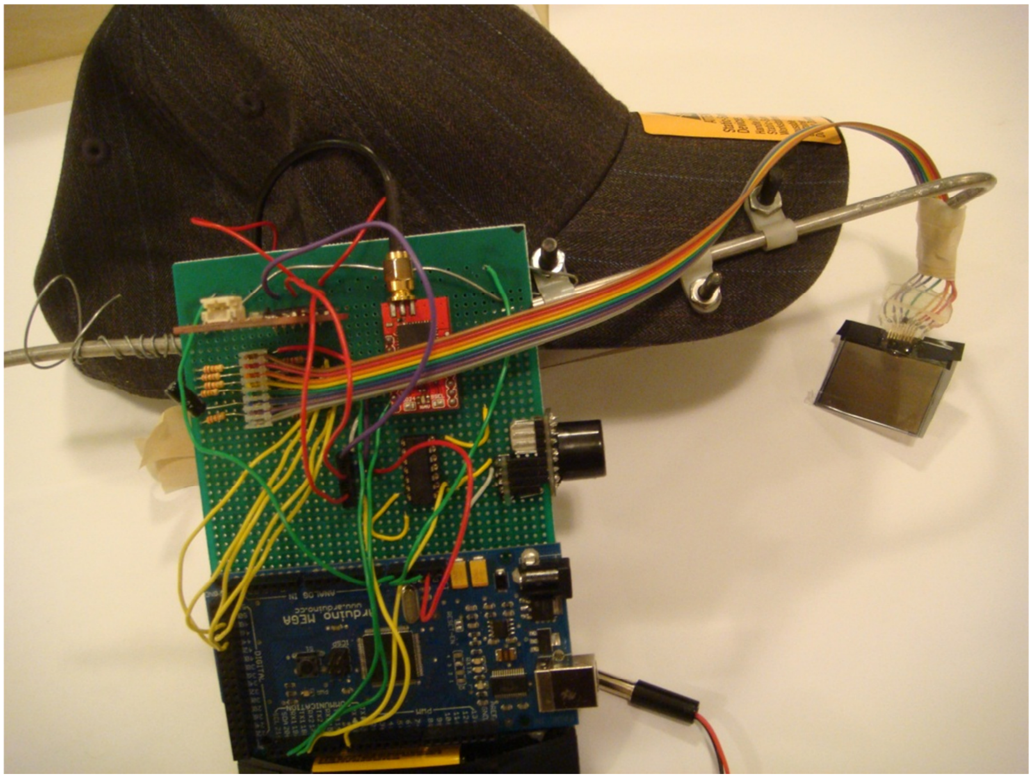

Figure 3: Physical Structure of the Heads-Up Display

The foundation of the Heads-Up Display is a simple baseball cap-style hat. The screen and electronics are attached to the hat by means of an aluminum rod attached to the right side of the hat, extending out in front of the wearer. The rod is attached to the hat with plastic brackets bolted to the brim of the hat with #8-32 screws. The electronics board is attached with stiff wire to the aluminum rod such that the electronics board hangs next to the wearer's right ear in a vertical orientation.

The aluminum rod is bent appropriately so that the screen, which is attached to the forward end of the rod, hangs in the ideal viewing position. Care has been taken in positioning the screen such that it does not impair the functionality of the sonar rangefinder.

The screen is taped in place at the end of the aluminum rod. Its data wires, in the form of a ribbon cable, run along the rod to the electronics board, where they connect via headers. The electronics board holds the Arduino Mega 1280, the optional 9V battery power source, voltage dividers for 3.3V logic, a 7400 series inverter IC, and most of the project's sensors, which are all described below. The GPS antenna is taped to the top of the hat, and its pigtail cable runs directly to the GPS unit on the electronics board, where it connects via an SMA cable.

Hardware Design

The Heads-Up Display electronics consist of the following:

- An Arduino Mega (ATmega 1280)
- A Sparkfun Razor 9 Degree-of-Freedom Inertial Measurement Unit
- A Sparkfun Venus GPS board
- A Sparkfun XL-Maxsonar EZ0 Ultrasonic Rangefinder
- A Sparkfun SMA GPS ceramic patch antenna
- A 7406 TTL hex inverter IC
- A Nokia 3310 monochrome LCD screen, 84x48 pixels
- Voltage divider resisters for the LCD signals
- An optional 9V battery for supplying power to the system (otherwise powered by USB)

The EZ0 rangefinder, the Venus GPS, and the Razor IMU are connected to the Arduino Mega via UART serial connections. The LCD screen is connected to the Arduino Mega using SPI. The EZ0 rangefinder's data signal is passed through one of the inverters on the 7406 chip to correct the active-low signal for use with the Arduino UART. The 5V SPI outputs for the 3310 LCD are each divided using 2k and 1k resistors to shift the voltage to 3.3V as required by the LCD's onboard circuitry. The entire system is powered through the Arduino Mega. The Mega itself can be powered through its USB connector (5V) or by a 9V battery connected into the 5mm barrel jack onboard the Mega.

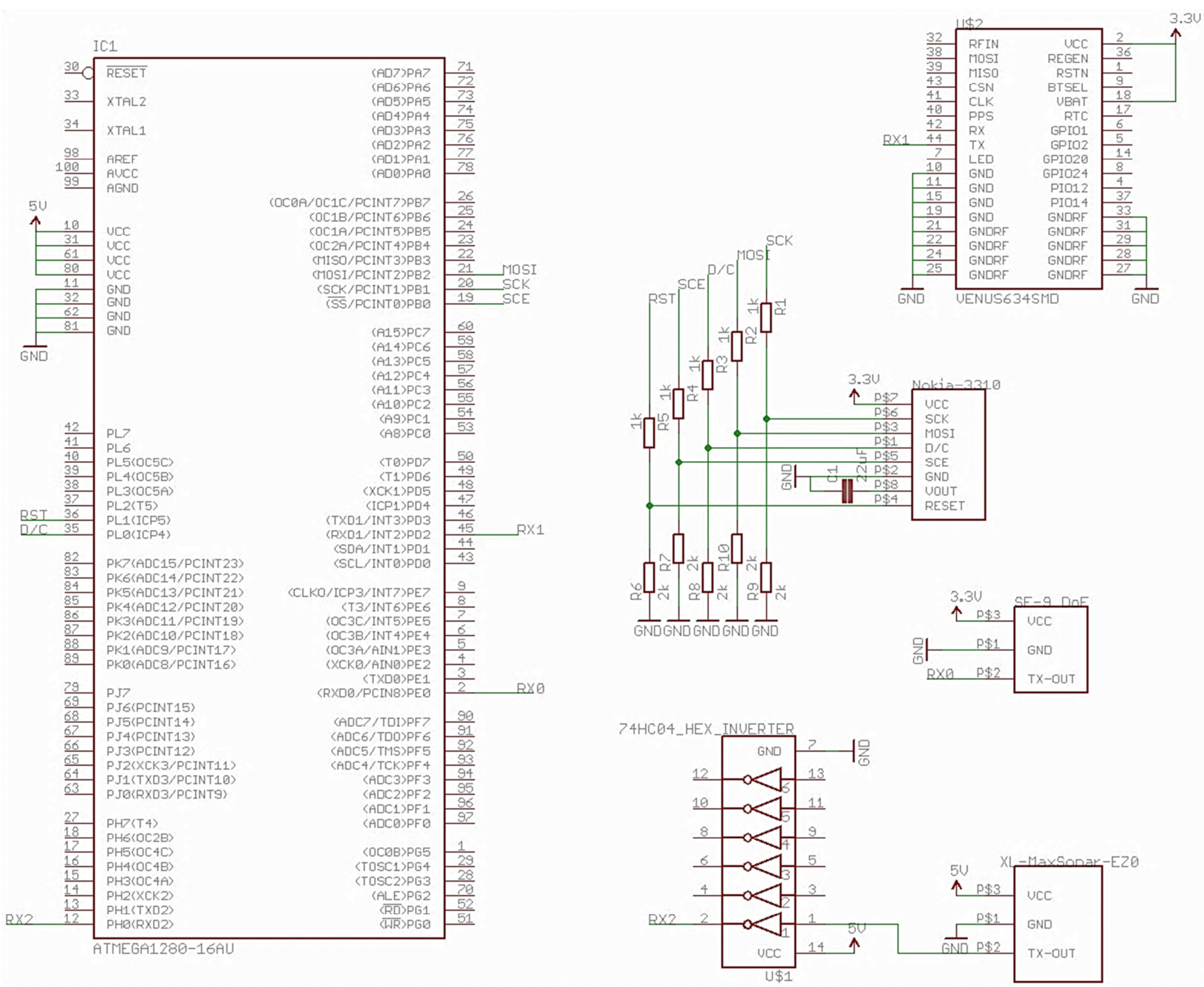

Figure 4: Hardware Schematic

Software Design

The control of the hardware was implemented in C code on the AVR, running the uCOS-II RTOS. Separate tasks were used for each hardware component that was implemented. Three control structures were used: a TIME control structure, a GPS location control structure, and a HUD state control structure. The TIME control structure holds, in ASCII characters, the hours and minutes corresponding to the current time read from the GPS. The GPS location data structures hold the latitude and longitude of both the current GPS position as well as the GPS destination. The HUD state control structure contains the TIME structure, the GPS location structures for both the current location as well as the destination, the distances, in feet, between the location and the destination in the N/S and E/W directions, the current heading in N,S,E,W, and finally the heading to travel to reach the destination in N/S and E/W.

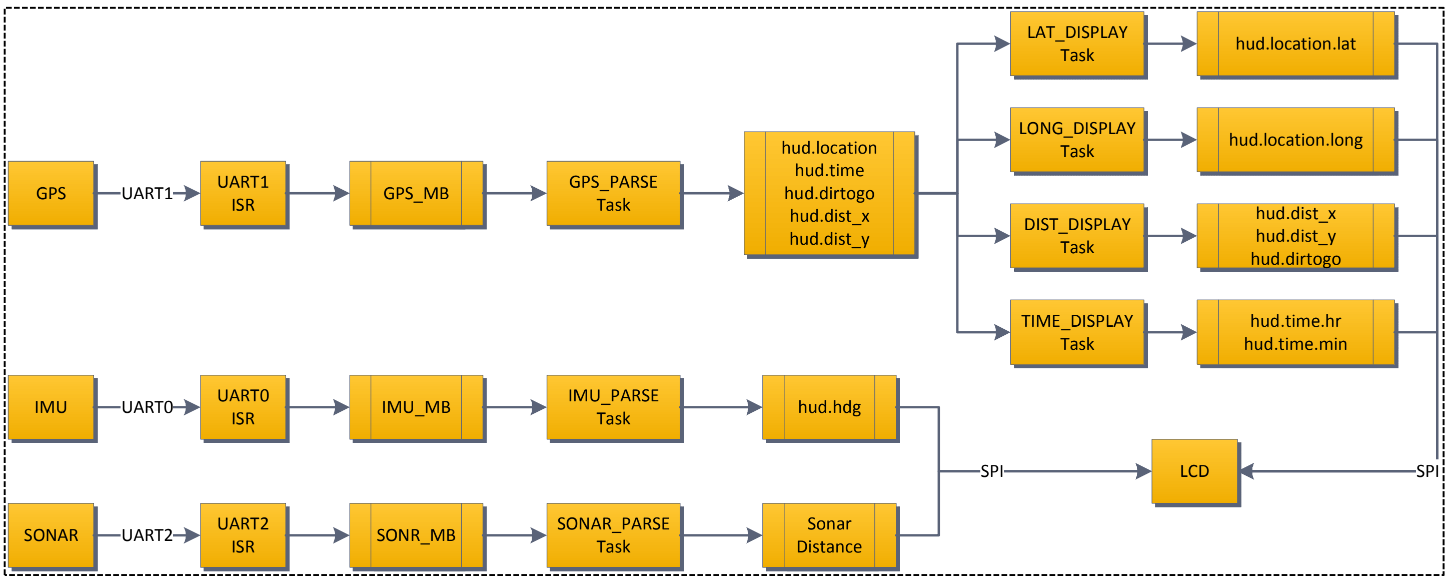

Figure 5: DFD for Heads-Up Display

The three UARTs used each had their own buffer into which the UART receive ISRs wrote the characters. When the end of line character is reached (0x0A), the ISRs sent the buffer out to each respective task through separate mailboxes. After sending the buffer through the mailbox, the ISR would reset and wait for more data from the UART.

The UART0 ISR wrote into the IMU mailbox, imu_mb, on which the IMU_PARSE task pended. The IMU_PARSE task checked the data in the mailbox to ensure that it was a properly formatted string, and then updated the heading of the HUD as indicated by the message. The IMU_PARSE task then displayed this heading on the LCD.

The UART1 ISR wrote into the GPS mailbox, gps_mb, on which the GPS_PARSE task pended. The GPS_PARSE task checked the data in the mailbox to ensure that it was a properly formatted string, and then updated the variables read from the GPS. The time was first read, formatted to be CST from GMT (by subtracting 6 hours) and then written into the HUD state data structure. The current GPS position, in degrees, minutes, and 10-thousandths of minutes, is read in, updated into the HUD state, and then the distance between current position and destination is calculated and stored in the HUD state as well.

The UART2 ISR wrote into the SONAR mailbox, sonar_mb, on which the SONAR_PARSE task pended. The SONAR_PARSE task checked the data in the mailbox to ensure that it was a properly formatted string, and then displays the distance on the screen.

There are four display tasks that periodically update different data on the screen. The TIME_DISPLAY task updates the time from the HUD state onto the screen. The LAT_DISPLAY and LONG_DISPLAY tasks update the current GPS position on the screen to the hundredths of minutes resolution. The DIST_DISPLAY task displays the distance to the destination in feet for both the N/S direction, and the E/W direction.

Documentation Listing

1. Technical Manual (this document)
2. User's Manual
3. Personnel Report
4. Datasheets & Schematics
    a. ATMEL ATmega Series Datasheet
    b. Arduino MEGA Schematic
    c. XL-MaxSonar Sonar Sensor Datasheet
    d. Venus GPS Datasheet
    e. Razor 9 DoF IMU Schematic
    f. 7406 TTL Hex Inverter