CSE 316 Project Report

A Smart Stick for Blind People
with object detection and direction guidance

Prepared By:
Sihat Afnan - 1705098
Nazmul Takbir - 1705103
Saif Ahmed Khan - 1705110

Supervised By:
Md. Tareq Mahmood
Lecturer, BUET CSE

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology

Contents

List of Figures

1 Introduction

The CSE 316 project scope was to create an embedded system with the Atmega32 microcontroller interfaced with various sensors to perform specific tasks. For our term project, we chose to build **SmartStick**, a stick for the blind with an embedded system mounted on it for obstacle detection and direction guidance.

2 Project Goal

Our goal was to develop an embedded system on a blind man's walking stick that would detect obstacles and guide the user via voice commands on an Android app using Bluetooth connectivity. The stick we made detects obstacles in the forward direction and instructs the user to turn left or right, wherever there is more free space.

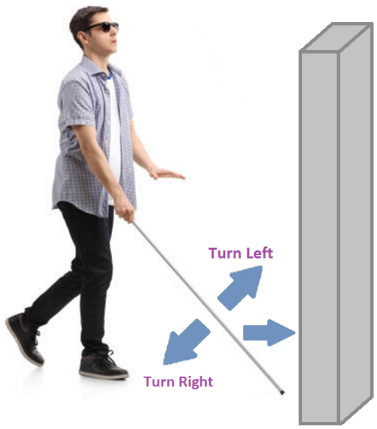

Figure 1: Scenario depicting usage

3 Components Used

3.1 Hardware Components

Component Name	**Quantity**
AtMega32	2
HC-SR04 Ultrasonic Sensor	3
HC-05 Bluetooth Module	1
7805 Voltage Regulator IC	2
9V Battery	2
Bread Board	2
USBasp programmer	1
Male to Male Wires	Many
Male to Female Wires	Many
Multimeter	2
Battery Connectors	2
IC Extractor	1

3.2 Software Components

Software Name	**Usage**
Android Studio	Developing the Android App
Extreme Burner	Burn .HEX file onto AtMega32
Atmel Studio	Writing Code for AtMega32

4 Block Diagram and Functionality

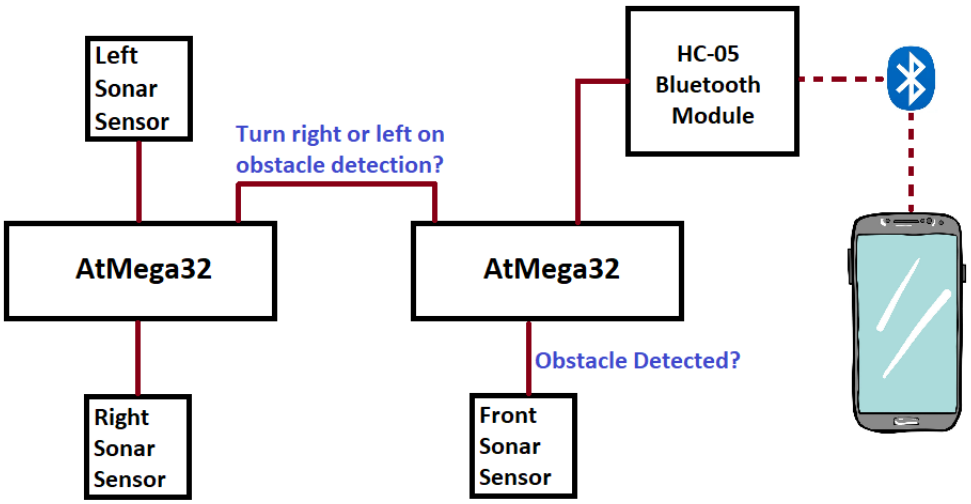

Figure 2: Block Diagram

4.1 How the system works

Figure 2 shows the block diagram of the embedded system. The AtMega32 on the left uses the two sonars interfaced with it to constantly measure which direction, left or right, has more free space. A single bit denoting turn left or right is sent to the AtMega32 on the right.

The right AtMega32 detects an obstacle in the forward direction and sends the signal to turn left or right using the HC-05 module to an Android app. The app we made receives this signal and plays the corresponding voice commands "turn left" or "turn right". If there is no obstacle in the forward direction, the system guides the user to keep going forward.

4.2 The Need for Two AtMega32

Ultrasonic sensors used in this project work by emitting sound waves when the Trigger pin is made high for 10us. When the sound wave is emitted, Echo pin of the ultrasonic sensor becomes high from low; when the reflected sound is received, the Echo pin becomes low again. So the time for which the Echo pin was high is the time for which the sound wave travelled. Using this value of time and the speed of sound wave in air, we can calculate the distance of the obstacle from which the sound wave was reflected back.

We need to find the time between a rising edge and falling edge of the Echo pin of the ultrasonic sensor. This can be done by connecting the Echo pin of the ultrasonic sonic sensor to an external interrupt of an AtMega32. The external interrupt starts a

timer when it detects a rising edge and stops the timer when it detects a falling edge. Therefore, we need to operate the external interrupt in a mode where it is triggered by both rising edge and falling edge. An AtMega32 has two external interrupts which can operate in such a mode - INT0 and INT1. However, we needed 3 ultrasonic sensors in our project. Hence, we used two AtMega32.

5 Circuit Diagram with Pin Connections

The detailed circuit diagram of the system with pinout is given in Figure 3. The link for the code of the two AtMega32 is given in Section 7. In the code, the left side AtMega32 of Figure 3 is referred to as ATMEGA 1 and the right side of Figure 3 is referred to as ATMEGA 2.

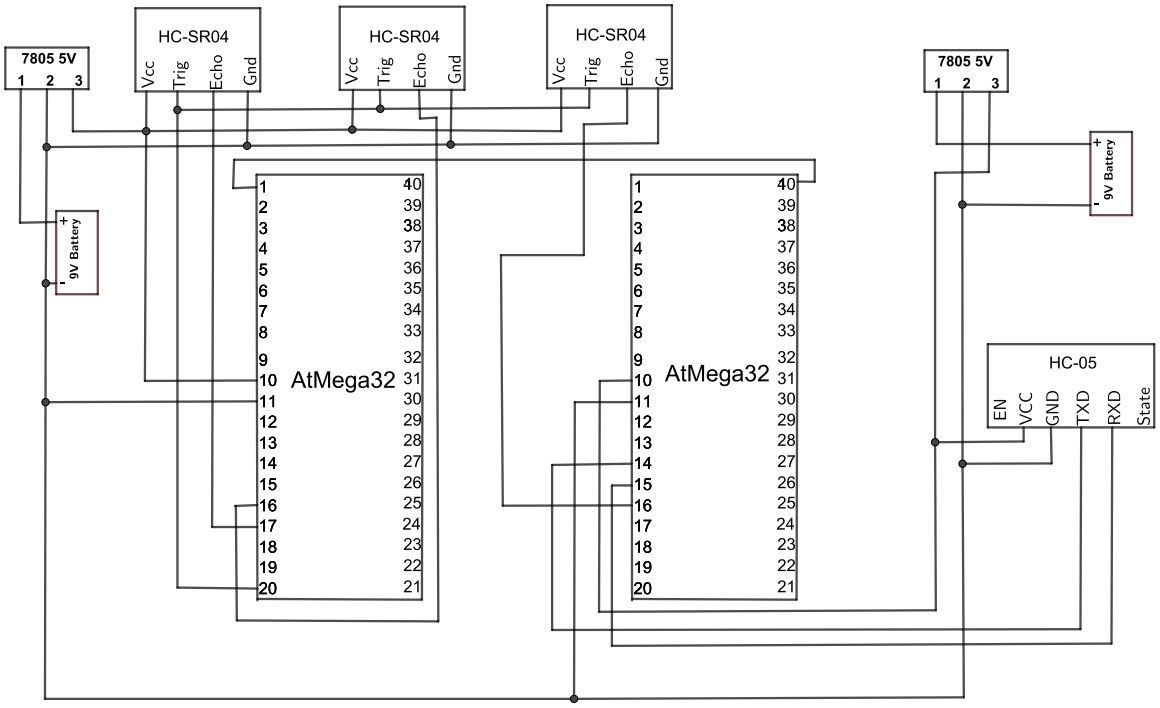

Figure 3: Circuit Diagram with Pinout

6 Difficulties Faced and their Solutions

- At first we tried to power the entire project using 1 battery, but the HC-05 module was not working. It was maybe due to insufficient current flow. Hence we used two 9V batteries.
- The 7805 IC does not produce 5V reliably if the input voltage is less than 7V. This is a problem because the voltage output of the 9V batteries decrease overtime. So we always kept some backup batteries while working.
- Faulty equipment was a major problem for us. Our first HC-05 module did not work properly, so we had to buy another. One of the breadboards we bought did not work well. Some of the wires we bought were faulty. Proper usage of the multimeter was essential in locating the source of problems when they arised.
- All the ground nodes in the circuit should be shorted.
- We had to use a Baud rate of 9600 for the USART communication through the bluetooth module. Double speed transmission mode should be used in this scenario, otherwise our module was transmitting garbage values.
- Nested interrupt is turned off by default in AtMega32. To turn it on sei() command should be used at the beginning of the Interrupt Service Routine.
- It is always a good practise to interface each sensor one at a time to see if it is working properly with the microcontroller.
- A nice trick to ensure successful burning of the code onto the microcontroller is to add an extra code to blink an LED on an unused pin (invert that bit inside the loop).
- Always use an IC Extractor to pull the microcontroller off the breadboard to avoid breaking the pins.

7 Resources

Links to the resources are given below:

- Code for the two AtMega32
- Code for the Android App
- Project Presentation