Alvin Tu
Nick Ong

Spider Sense Final Project Proposal

Project Site

https://sites.google.com/view/spideysense/

Overview/Goal/Purpose

Our final project draws inspiration from one of the most beloved superheroes: Spider-man. We plan to construct a mask/helmet that will simulate his "Spidey-Sense" ability and to allow users to determine which direction a potential hazard is coming from. LiDAR sensors on the side and back of the helmet will alert the user of any incoming objects that are not in their field of view, and a vibration motor from the same direction as the object will notify the user of its presence. Additionally, there will be a bluetooth display that connects to another device to show the stats such as the direction or distance of the incoming objects.

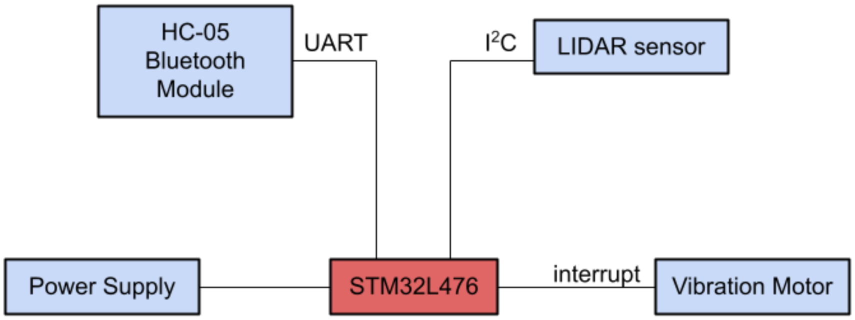
Peripherals

LiDAR sensor, HC-05 Bluetooth module, vibration motor

Serial Interface Protocols

I^2C (LiDAR), UART (Bluetooth)

Block Diagram



Responsibility List

Alvin:

- Configuring LiDAR with I2C to detect distances for vibration motor
- Procuring parts, implementing algorithm for distance detection
- Implement STM board with a power supply (so we can make it mobile)

Nick:

- Configuring HC-05 Bluetooth module with UART to work w/ Termite
- Develop the interrupt code for the vibration motor
- Implement speed detection algorithm

Both:

- Build helmet/mask

Software

1. Initialize GPIO Pins (AF for I2C and UART, Input/Output, Moder, etc.)

2. Initialize I2C for STM board and LiDAR sensor (STM = Master, LiDAR = Peripheral)

3. Initialize and configure UART to set up communication with HC-05 Bluetooth Module

4. Initialize an interrupt for when an object gets too close as detected by the LiDAR.

5. Set up the interrupt handler for when an object gets too close as detected by the LiDAR (vibration motor will vibrate)

6. Configure main loop to display stats of Spidey-sense through Bluetooth communications with phone or other device

    - The main file will call all of our initializing functions. Then, there will be a while loop to continuously check and update volatile variables. There will be conditionals to check if an object gets too close, and if it does, an interrupt will trigger the vibration motor.