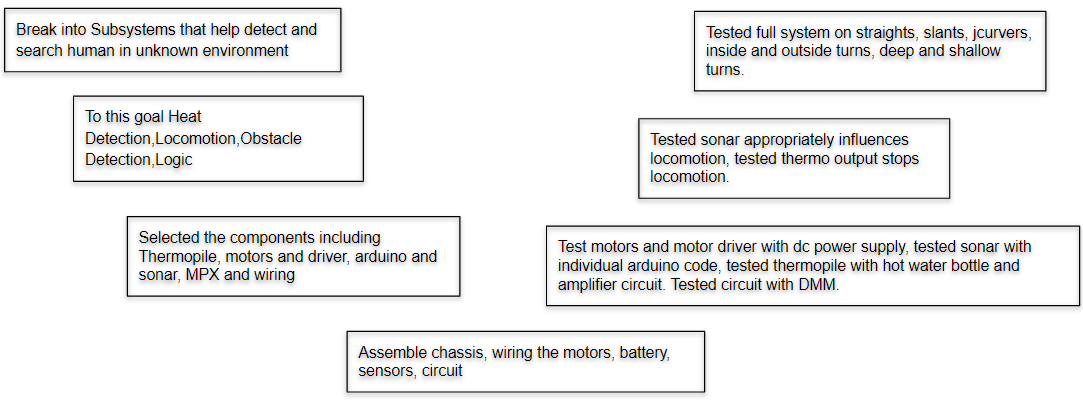
**Team 6 Schematics and Validation: Richard Wan, Kevin Zheng, Sam Park, Sartaj Chowdhury**

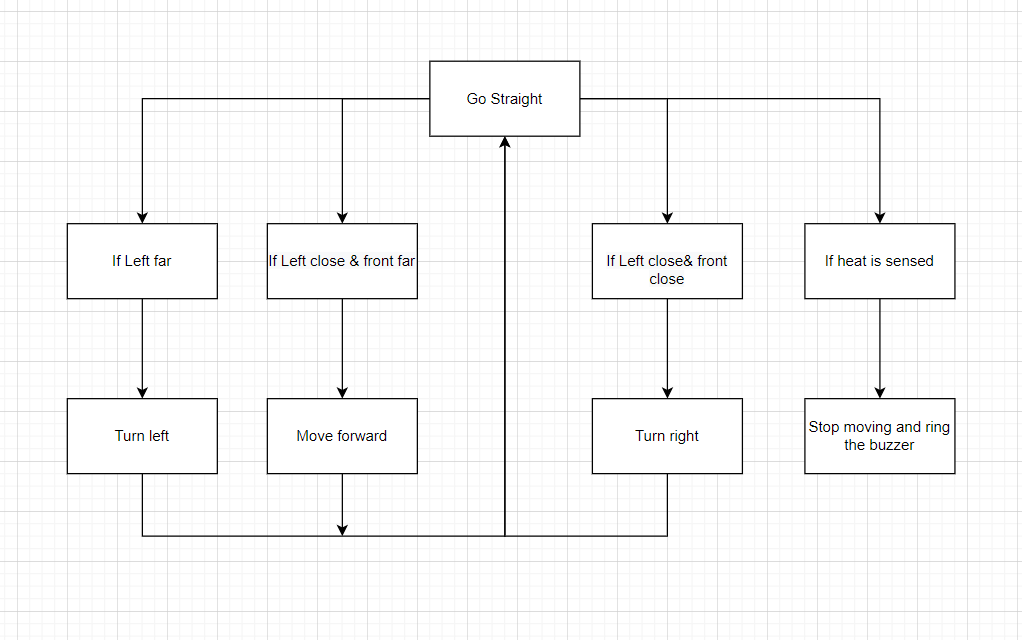
**V Diagram**

****

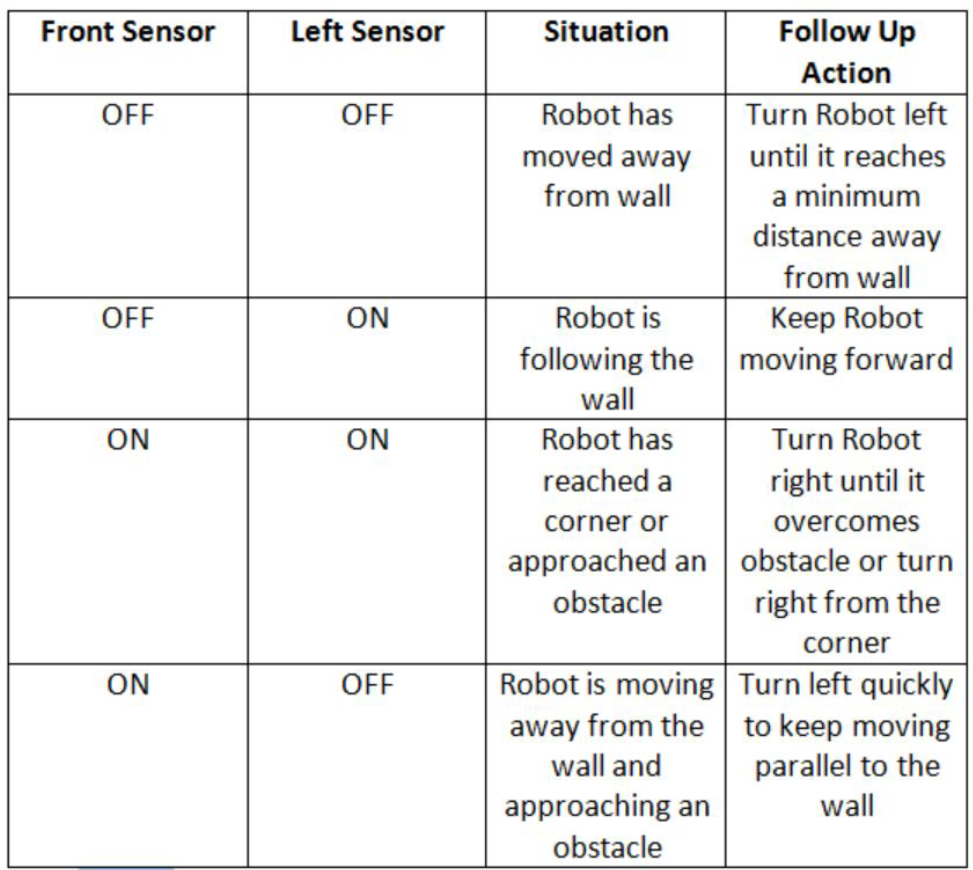
**Logic**

1. The front and left sensor send data to the arduino
2. Depending on the sensor distance, determine if the two sensors are close or far
3. If the left sensor is far, turn left and move forward a little
4. Left sensor is close and
   1. If front sensor is close, turn right
   2. If front sensor is far, go straight with a bit of right turn biase
5. If the thermophile detects a hot object, set the pwm to 0 for both wheels, send 5V to the buzzer, and delay indefinitely.

**Robot and code flow chart:**



**Truth tables:**

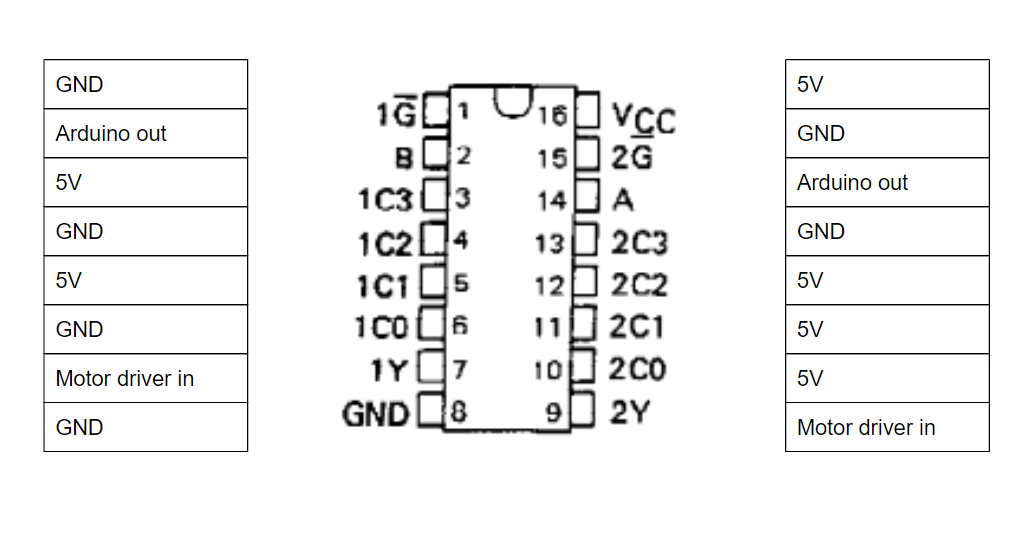


| SF | SL | I1L | I2L | I1R | I2R |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |

Input Assignments

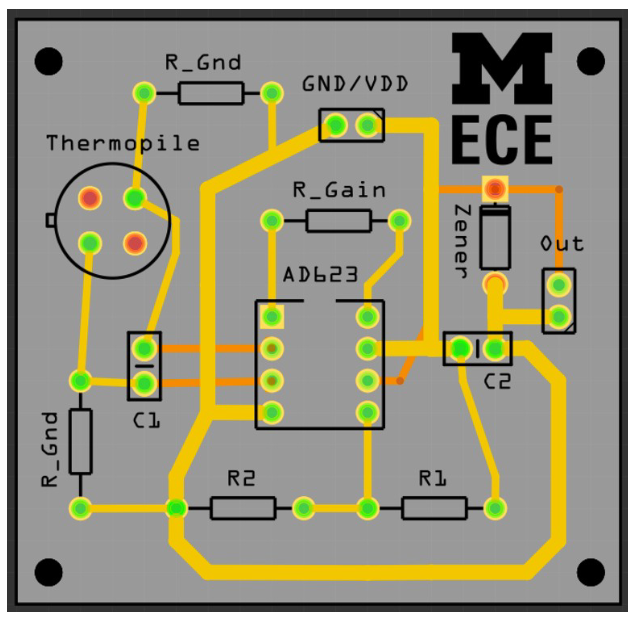
| SF | SL | I1L | I2L | I1R | I2R |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 | C0 = 0 | 0 | 0 | C0 = 1 |
| 0 | 1 | C1 = 1 | 0 | 0 | C1 = 1 |
| 1 | 0 | C2 = 0 | 0 | 0 | C2 = 1 |
| 1 | 1 | C3 = 1 | 0 | 0 | C3 = 0 |

There are only four conditions for a left wall following algorithm and there are only 3 behaviors for the robot to exhibit. These include “front/left:far/close” and behaviors “turn left/right” or “go straight”. Using this knowledge we simplified our wiring as the motor driver uses two binary inputs to control wheel rotation. For the 3 behaviors, the wheels will never have to rotate backwards which greatly reduces the complexity of the situation. Thus certain inputs on the motor driver can remain grounded as they will not need any input from the multiplexer. Therefore assigning the multiplexor selection pins to the front and left sensors we can adapt the wall following algorithm truth table to multiplexor and motor driver truth tables. These are shown above. Leaving the permanently grounded ports on the motor driver we assigned the two outputs on the multiplexer to the remaining 2 inputs on the motor driver. The enable pins were permanently grounded as the locomotion termination was done through arduino pwm output as specifically allotted in lecture.

**Multiplexer wiring:** ****

**Heat Detection Subsystem:**

**Thermopile Circuit diagram**

****

This amplifier circuit is powered by our 5 volt rail on the breadboard and amplifies the signal from our thermopile sensor and outputs the amplified voltage into the arduino analogread(). This allows us to set a threshold voltage to detect ranges of temperature and the value we chose to detect human body temperature was 270 on the analogread(). Once a value higher than 270 is detected by the pin, the PWM pins to the motor driver output 0 to stop the motors and the buzzer pin sends 5V to the buzzer to ring it. This proved very flexible and was able to be tuned more sensitively depending on the heat of the target. This was made extra sensitive for the final test where it successfully detected the bottle. The heat detection system was completed using a circuit from the appropriate lab and carried over into the adapted arduino iteration used on the robot. Testing was done using the DMM and detecting high voltage output individually from the circuit output. The next verification step was allowing the robot to use the circuit on its own to detect hot objects. And was attached using tape to the front of the robot and wired into the 5V rail of our breadboard for powering the opamp as well as direct output into the arduino to transmit its information.