How it's Coded

Code:
[Itz CODE](#)

What lines DO

Line 1: Sets up the keyboard, GPIO, And time for later use in the code

Lines 3-4: defines values for Echo and Trig for sonar

Lines 5-10: sets up GPIO board to power the motors

Lines 11-18: sets up the Sonar Gpio slots

Lines 19-20: Shuts off GPIO so the motors don't start on

Lines 22-26: defines the camera ON function

Lines 28-30: defines the camera OFF function

Lines 32-47: defines sonar function

Lines 49-54: defines backwards functions

Lines 56-61: defines forwards function

Lines 63-74: Defines left turn function (Note that the turn has has two settings for higher and lower speeds)

Lines 76-87: Defines right turn function (Again note that the turn has has two settings for higher and lower speeds)

Lines 89-94: Defines stop function

Lines 96-97: Starts active code

Lines 98-99: Does a sort of emergency shutdown when Q is pressed

Lines 101-102: Activates sonar function

Lines 104-119: Takes output of sonar function and goes forward backwards or turns left depending on how close an obstacle is (Note this basically makes the robot go forward unless it sees a wall) when W is pressed

Lines 111-112: Starts the backwards function when S is pressed

Lines 114-115: Starts the right turn function when D is pressed

Lines 117-118: Starts the left turn function when A is pressed

Lines 120-124: Takes power from motors when no keys are pressed

Lines 126-127: Starts recording when V is pressed

Lines 129-130: Stops recording when C is pressed

Lines 132-160: Sets Motor power to a percentage when the corresponding key is pressed ex. Power set to 70% when 7 is pressed

162-163: Shuts down code and cuts power to GPIO

Wiring

Baby Shark do do do do do do Baby Shark do do do do do do Baby Shark

- First you need to connect a keyboard and mouse to the Pi using the USB ports on the Pi.
- You also need to connect the Raspberry Pi to the monitor with an HDMI cable.
- To power the Pi you need to connect a micro usb cord into the Pi in the designated spot then plug the other side into an outlet.

LED SetUp:
To use LEDs on with the Pi, you will need a breadboard along with resistors, the Pi itself, and of course, the LEDs.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
GPIO.setup(7,GPIO.OUT)
GPIO.setup(11,GPIO.OUT)
GPIO.setup(13,GPIO.OUT)
GPIO.setup(15,GPIO.OUT)
GPIO.output(7,True)
time.sleep(1)
GPIO.output(7,False)
GPIO.output(11,True)
time.sleep(1)
GPIO.output(11,False)
GPIO.output(13,True)
time.sleep(1)
GPIO.output(13,False)
GPIO.output(15,True)
time.sleep(1)
GPIO.output(15,False)
GPIO.cleanup()
```
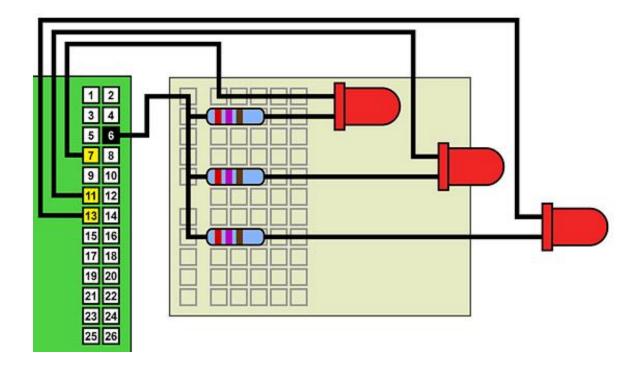
LED Lights Wiring:
> You will have to connect the pins with the holes in the breadboard using jumpers.
> Use the six pin to connect resistors to each of the lights.
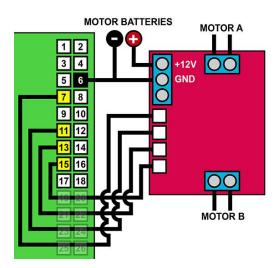> Then use jumpers to connect the breadboard to the GPIO pins on the Pi.
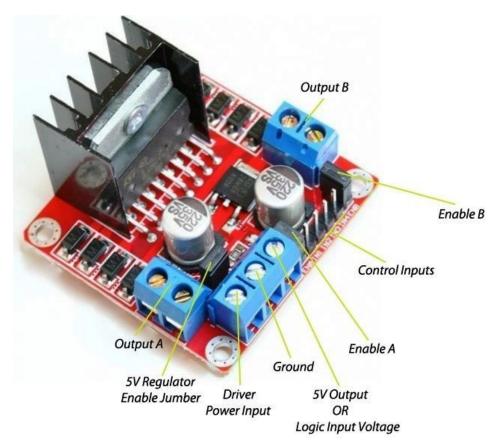
Motor Wiring:

In the picture below, pins 7 and 11 control motor A, and pins 13 and 15 control motor B.

Pin 6 is the power / battery connection. So, 15,7,11, and 13 all connect into the control inputs on the H-Bridge.

Ultrasonic Sensor:

```python
def
ultrasonic()
:
        #Gnd wire (Green)connects to breadboard/ connects to wire (Black)14

        GPIO.output(TRIG, True)                #Set TRIG as HIGH

        time.sleep(0.00001)                    #Delay of 0.00001
    seconds
        GPIO.output(TRIG, False)               #Set TRIG as LOW
```
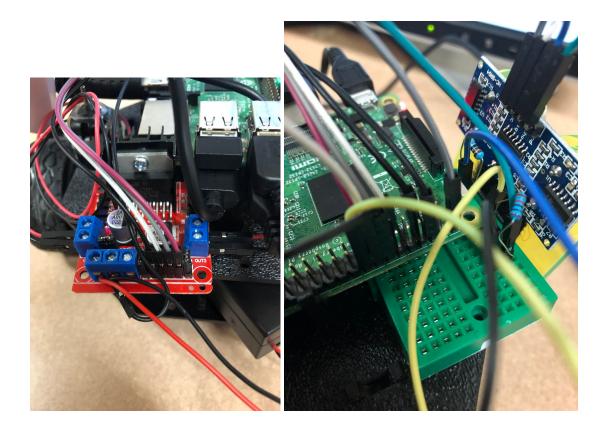
```python
    while GPIO.input(ECHO)==0:                    #Check whether the
ECHO is LOW
            pulse_start = time.time()             #Saves the last
known time of LOW pulse


    while GPIO.input(ECHO)==1:                    #Check whether the
ECHO is HIGH
            pulse_end = time.time()               #Saves the last
known time of HIGH pulse


    pulse_duration = pulse_end - pulse_start #Get pulse duration to
a variable


    distance = pulse_duration * 17150        #Multiply pulse
duration by 17150 to get distance
    distance = round(distance, 2)            #Round to two decimal
points


    return(distance);
```

uses the ultrasonic sensor to shoot out a sonic pulse then receive
the pulse to tell the distance between the robot and the object in
the way.

Camera code:

https://github.com/frc2052/MiniRobot/blob/648cbad5f78a3bc5750277e9dd4426d8176b639f/camera.py

Credits:

https://www.explainingcomputers.com/rasp_pi_robotics.html