*Sten***_BOT_** Robot Kit

Sensing

Legal Stuff

- Stensat Group LLC assumes no responsibility and/or liability for the use of the kit and documentation.
- There is a 90 day warranty for the Quad-Bot kit against component defects. Damage caused by the user or owner is not covered.
  - Warranty does not cover such things as over tightening nuts on standoffs to the point of breaking off the standoff threads, breaking wires off the motors, causing shorts to damage components, powering the motor driver backwards, plugging the power input into an AC outlet, applying more than 9 volts to the power input, dropping the kit, kicking the kit, throwing the kit in fits of rage, unforeseen damage caused by the user/owner or any other method of destruction.
- If you do cause damage, we can sell you replacement parts or you can get most replacement parts from online hardware distributors.
- This document can be copied and printed and used by individuals who bought the kit, classroom use, summer camp use, and anywhere the kit is used. Stealing and using this document for profit is not allowed.
- If you need to contact us, go to www.stensat.org and click on contact us.

References

- www.arduino.cc
- https://github.com/esp8266/Arduino

Robot Sensing

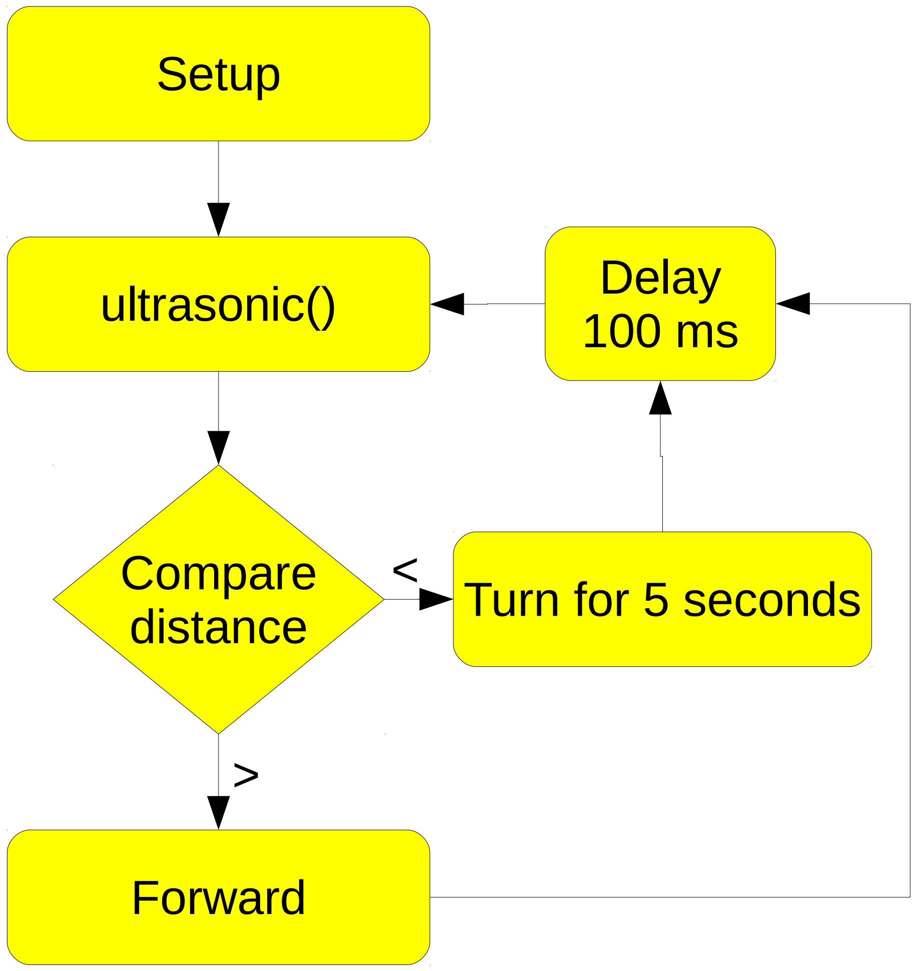
- This section, you learn about using sensors to control the robots movements.

Caution

- Do not use D0 and D2 as inputs. They can be used as outputs.
- At power up or reset, the SLATE processor uses D0 and D2 to determine how to boot. Both pins need to be in the high state. There are resistors that pull the pins to 3.3 volts, the high state. When connecting to the pins, make sure the pins are not pulled low at power up. This will keep the processor from properly booting and operating.

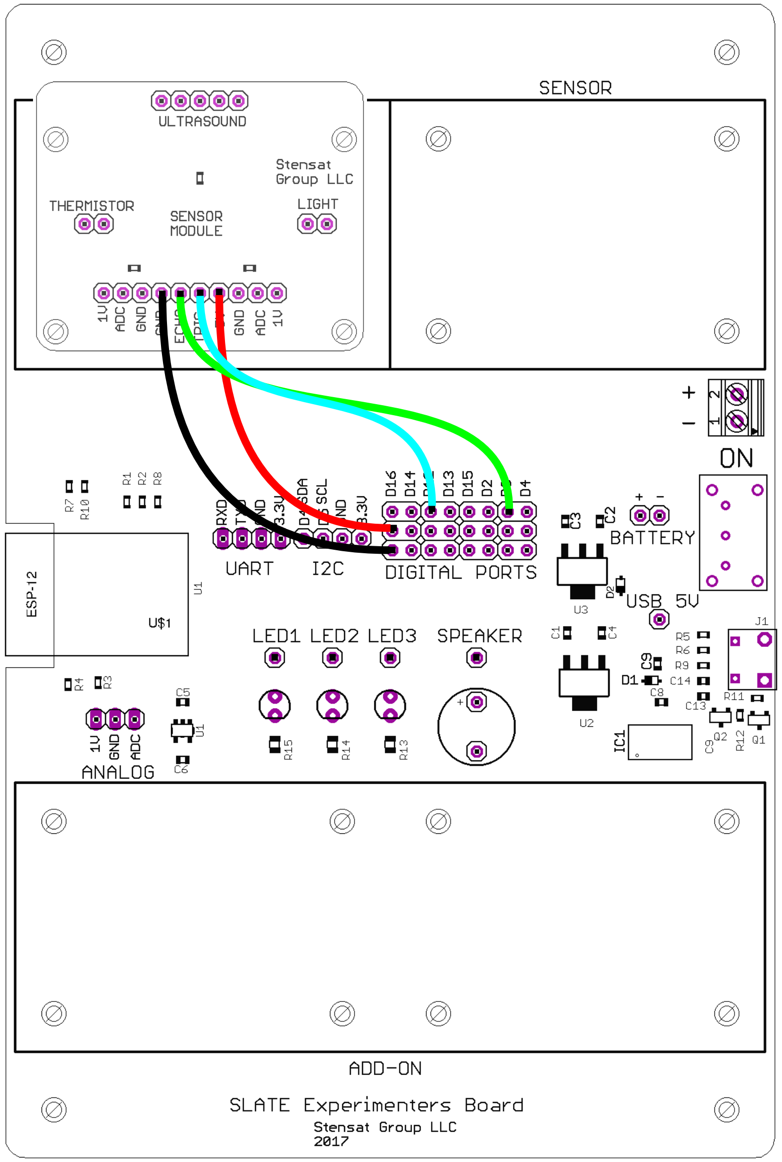
Ultrasonic Range Sensor

- The ultrasonic range sensor will be used for collision avoidance. The program operates similar to the photo cell program.
- Reuse the ultrasonic function from the SLATE lessons. It will take the place of the analogRead() used for the photo cell.
- Pick a distance as the threshold for comparison.
- Notice that the robot moves forward as long as the distance is greater than the threshold otherwise turn.



Ultrasonic Range Sensor

- Install the sensor module in the sensor area.
- Connect the wires as shown.
  - Connect the sensor **GND** to ground on a digital pin.
  - Connect the sensor **5V** to a center pin on a digital pin. **D16** is used here.
  - Connect the **ECHO** signal to **D0**.
  - Connect the **TRIG** signal to **D12**.



Ultrasonic Sensor

- Modify the ultrasonic function from the SLATE lessons to use the new digital pins.

```
long ultrasonic()
{
    digitalWrite(12,LOW);
    delayMicroseconds(2);
    digitalWrite(12,HIGH);
    delayMicroseconds(10);
    digitalWrite(12,LOW);
    long distance = pulseIn(0,HIGH);
    if(distance == 0)
        return(1000);
    distance = distance/58;
    return(distance);
}
```

Collision Avoidance Program

- The program on the next page will use the code used to control the motors, the ultrasonic function, and the conditional command.
- Put together, the program will keep the robot from bumping into anything.
- Enter the code on the next page. The code should be written in a single file. The code is split on the next page since it wouldn't fit in a single column.
- You will notice a **`delay()`** at the end of the **`loop()`** function. This is needed because the ultrasonic range sensor cannot be operated too fast. Incorrect results will occur if the loop runs too fast.
- Test it and see if you need to tweak the timing for going reverse and turning.
- Don't forget to include the motion file by adding the file.
- Save the program and then upload it.
- Change the code to turn a different direction.

Collision Avoidance Program

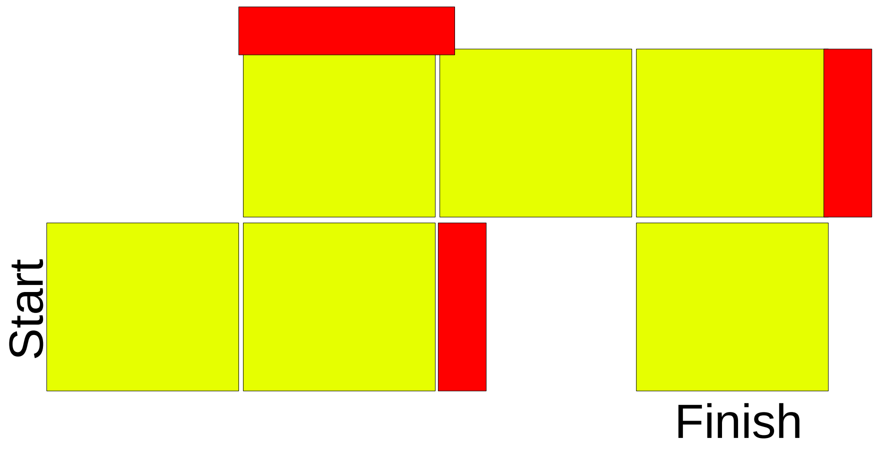
```
long ultrasonic()
{
    digitalWrite(12,LOW);
    delayMicroseconds(2);
    digitalWrite(12,HIGH);
    delayMicroseconds(10);
    digitalWrite(12,LOW);
    long distance = pulseIn(0,HIGH);
    if(distance == 0)
        return(1000);
    distance = distance/58;
    return(distance);
}

void setup()
{
    pinMode(0,INPUT);
    pinMode(15,OUTPUT);
    pinMode(16,OUTPUT);
    pinMode(14,OUTPUT);
    pinMode(13,OUTPUT);
    pinMode(12,OUTPUT);
}
```

```
void loop()
{
long distance;
    forward();
    distance = ultrasonic();
    if(distance < 10) {
        reverse();
        delay(1000);
        left();
        delay(700);
        halt();
    }
    delay(50);
}
```

Obstacle Course Time

- Now for the fun part. Modify and expand the program to go through the obstacle course shown below. The large square represent 2 foot grids. The red rectangles represent a barrier that can be detected with the ultrasonic range sensor. Set up some barriers out of any solid material. Card board boxes, poster paper, or other large materials will work.
- Use the ultrasonic range sensor to avoid crashing into the barriers and turns the right direction every time a barrier is detected.
- Hint, use the collision avoidance program and expand it so that it will complete the maze. This requires the robot to back up and turn in specific directions at specific points of the maze.



Stensat Group LLC