What shall we do today?

1. We were able to light an LED the previous day. So, let's do something more useful today.
2. We will be making a simple alarm system.

Getting started.

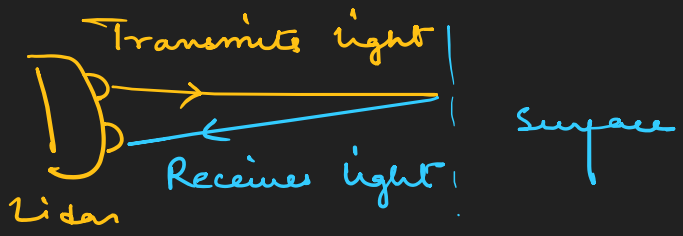
1. Open a new Arduino IDE sketch and save it with any name you want. I will be using alarm_system.
2. As discussed, it will create a new folder named alarm_system and a new ino file inside it.
3. Now we are good to go.

Materials required.

1. Sensors: A lidar, this is the distance sensor that we will be using to sense if someone is close to our system.
2. Actuators: A buzzer, this will make a loud beep when we detect someone with our lidar.
3. Microcontroller: Arduino UNO.
4. Miscellaneous: Bread board and jumper wires.

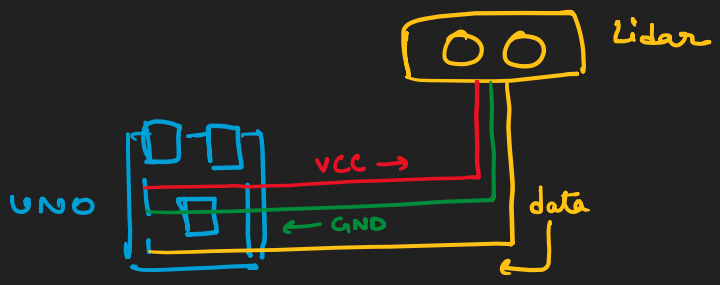
What is a lidar?

1. Lidar stands for light detection and ranging. They are sensors used to measure distance.
2. Lidars come in various forms and factors. We will be using the SHARP analog lidar. Visit https://robu.in/product/sharp-distance-measuring-sensor-unit-20-150-cm-gp2y0a02yk0f/ if you want to check more technical details about it.
3. The working of a lidar is similar to echo-location used by bats or sonars used by ships.



How can we read data from this particular lidar?

1. The SHARP lidar has 3 pins: VCC, GND and data.



2. Most sensors (this lidar for example) require electricity to run, and for electricity to flow we need a path. VCC and GND allow that.
3. VCC is connected to the positive terminal of a battery (or power source, UNO's 5V pin in our case) and GND is connected to the negative terminal (UNO's GND pin). VCC in most sensors is represented by red wire and GND is represented by a green or black wire.
4. Recall that analog IO can be used to send and receive 10-bit values. So the SHARP lidar sends an analog signal through the data wire which is then read by the UNO using analogRead().

Let's write a bit of code.

1. We start with the blank Arduino template.

```
void setup() {

}

void loop() {

}
```

2. We will be using A5 pin (any analog pin should be fine) for reading the analog data sent by the lidar. Since UNO is receiving data from lidar, it will be an INPUT pin. So we set A5 pin as an INPUT pin in the setup().

```
void setup() {
  pinMode(A5, INPUT);
}

void loop() {

}
```

   A5 is a predefined macro for UNO boards, we can just use it without worrying about what the real pin number for A5 is. You can hover over A5 to check the real pin number too. Like OUTPUT, INPUT is also a macro and is used to specify that a pin is to be used for taking inputs (hover over it to know its value).
3. Next step, we have to read the data sent by the lidar using analogRead(). We will read the data in loop(). Why? Because it is something we have to keep doing forever.

```
void setup() {
  pinMode(A5, INPUT);
}

void loop() {
  int data = analogRead(A5);
  delay(500);
}
```

   We used delay(500) so that UNO waits for half a second before checking the lidar again, just to make UNO work a bit less.

What is the unit of the data obtained?

1. We can't work with the raw data sent by the lidar.
2. It is not in any metric unit like cm or mm, so how do we know what it is trying to say?

3. Electronic devices come with a manual so that users can understand how to use them. The manual for our lidar is here: https://robu.in/wp-content/uploads/2016/03/SHARP-Distance-Measuring-Sensor-GP2Y0A02YK0F.pdf
4. Manuals are filled with overwhelming technical details, but let me point out some important things because learning to read manuals is a very important skill to become good with IoT.

can be read using analogRead()

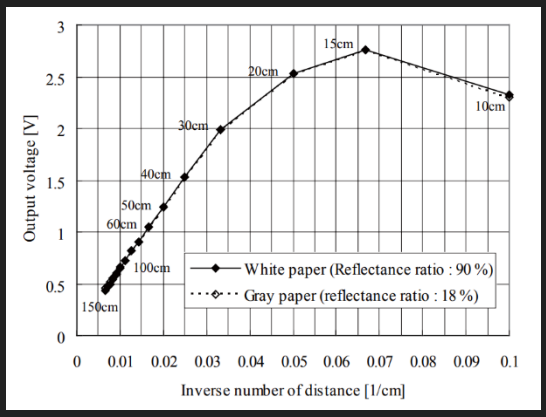
■Features
1. Distance measuring range : 20 to 150 cm
2. Analog output type
3. Package size : 29.5×13×21.6 mm
4. Consumption current : Typ. 33 mA
5. Supply voltage : 4.5 to 5.5 V

works if the object is between 20 cm and 150 cm.

can be powered by 5V pin

Note that current rating is also important. Our laptop USB ports can usually provide only up to 500mA, if our devices use any higher than that then it is not good for our laptop, so use an external power supply depending on your devices.

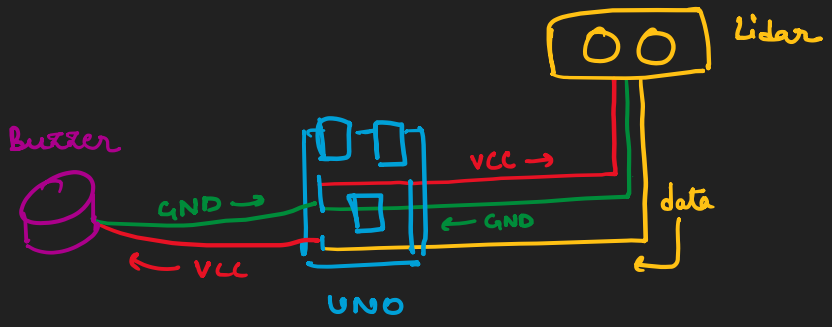
5. Now time for the great revelation:
   a. The manual provided us with a graph to depict the relation between distance and output voltage. Output voltage is the analog data sent by the lidar.
   b. We can infer from the graph that data is proportional to the inverse of distance for our range of 20cm - 150cm.
   c. If we want to set our alarm if an object is within 50cms, we can see from the graph that the output voltage should be higher than 1.25V. Since the total voltage range of 5V is divided into 1024 parts (recall the discussion on analog IO), 1.25V will correspond to 1.25/5 * 1024 = 256 (we will be using an approximate value of 260).
   d. It is also possible to find a mathematical function to convert analog data into cms, but I'll skip that part. You may explore that if you are confident with your maths knowledge, it's just straight lines (more or less).



Note: If you are not able to understand, you may jump to the next section just knowing analogRead() value should be greater than 260 if an object is less than 50cms away. But do try to re-read a few more times, it only uses high school maths and so that you can read the manuals of any other analog lidar you are provided with.

The buzzer.

1. The final thing left is to activate the buzzer.
2. It is like an LED, just send it a HIGH signal to its positive terminal and it will start beeping.
3. The more voltage it receives, the louder it beeps. So you can either use digitalWrite() or make it less loud by sending some low value of analogWrite().
4. So let us insert the negative terminal of the buzzer into another ground of the UNO and positive terminal into A0.



The final code.

1. We will continue from here.

```
void setup() {
  pinMode(A5, INPUT);
}

void loop() {
  int data = analogRead(A5);
  delay(500);
}
```

2. Now we have to set pin A0 as OUTPUT pin, because we will be using it to send voltage to the buzzer.

```
void setup() {
  pinMode(A5, INPUT);
  pinMode(A0, OUTPUT);
}

void loop() {
  int data = analogRead(A5);
  delay(500);
}
```

3. Last step: send some signal to the buzzer if data is greater than 260, else 0 will be sent.

```
void setup() {
  pinMode(A5, INPUT);
  pinMode(A0, OUTPUT);
}

void loop() {
  int data = analogRead(A5);
  if (data > 260) {
    analogWrite(A0, 500);
  } else {
    analogWrite(A0, 0);
  }
  delay(500);
}
```

We chose to send 500 to the buzzer, the maximum value is 1024; so it is less than 2.5V and it shouldn’t be too loud.