

Touch Bar

Team

Team Name- Beginners

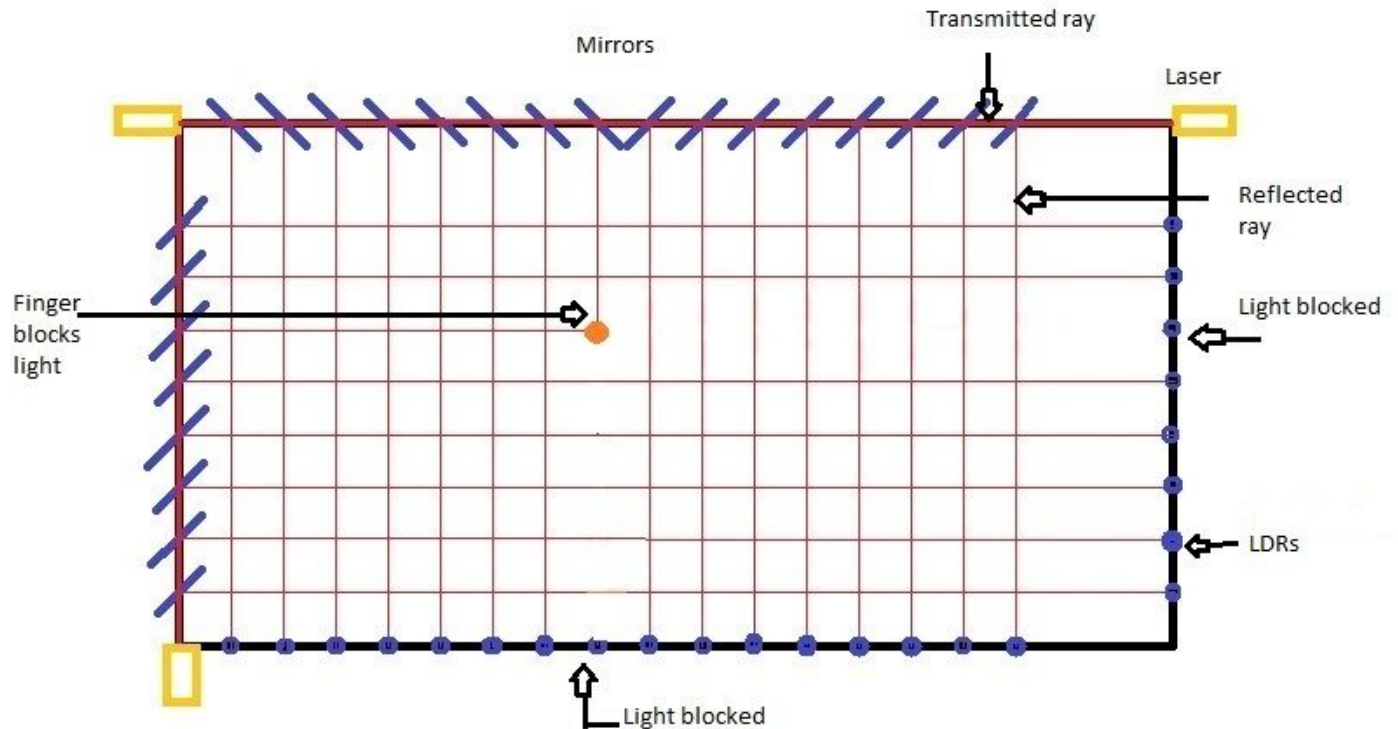
Name	Roll Number	Email	Mobile No
Shubham Shah	150100042	shubhamshah1997@gmail.com	9028378859
Nandan Prince	15D260008	nandanixr@gmail.com	8828290945
Deepak Singh	150260011	deepaksingh24081996@gmail.com	9685685527

Project

Our initial project was to make an external device that can be connected to the laptop through a cable and would enable touch sensitivity on screen. We later changed it to a frame over which if you move your hand, a corresponding shape is created on the laptop screen in the processing software.

How we did it

- 1) **Frame** : We used wooden strips to make a rectangle the size of a laptop screen and glued them together using Araldite.
- 2) **Making the grid**: So as to find the position of the user's finger, we had to make a virtual grid with light rays over the frame. Initially we thought of using LED's as a light source and photodiodes as receiver. But the spread of the rays from LED, caused interference at the corresponding receiver. So we thought of using lasers and +LDR, but using so many lasers would have been really costly. Finally, we used mirrors and did the job with just 3 lasers. The image represents our method. The mirrors were to be placed at exactly 45 degree angle and also needed to be adjusted so that the reflected ray falls directly on the LDRs and not above or below them. Mirrors and lasers were stuck using bondtite and feviquick. LDRs were stuck on a strip of double sided tape.



3) **Encoder circuit:** The signals from the LDRs were to be sent to the Arduino for processing. But we had 24 input signals and the Arduino didn't have as many pins. So, we used an encoder circuit which converts decimal to binary. We made two encoder circuits- one 16 to 4 converter and one 8 to 3 converter (for the horizontal and vertical rows of LDRs respectively.) Encoder circuit was available from the course material of EE112

Also, this was one of the reasons why we didn't cover the entire frame, as then the number of signals won't be a power of 2.

4) **Arduino processing:** Coding learnt from- <https://stab-iitb.org/electronics-club/tutorials/arduino/>

The output from the encoders were the binary representation of the LDRs that does not receive light hence providing the x-y coordinate of the touch. One problem we faced was that even when high, the voltage across the LDR's was around 2.5 volts which the arduino (digital pins) counted as low. So, we had to use two arduinos and used the analog inputs instead and took values above 400 as high.

To Arduino 1 were sent the three signals from the 8 to 3 converter and the four signals from the 16 to 4 converter were sent to Arduino 2.

Arduino 1 code- <https://github.com/Shubham-Shah/ITSP/blob/master/sender.ino>

Takes 3 analog inputs. Considers them high if value>400 and writes them as output f=on the digital pins. These pins are connected to three digital input pins on arduino 2.

Arduino 2 code- <https://github.com/Shubham-Shah/ITSP/blob/master/Receiver.ino>

Takes 4 analog inputs and 3 digital inputs(from Arduino 1). Converts them to decimal (e.g.- 1101 in binary. Decimal= $1*(2^3)+1*(2^2)+0*(2^1)+1*(2^0)=13$).

Hence we get x and y co-ordinates in terms of the number of LDRs.

This is used as Total= $x*10+y$ which is written on to the Serial monitor and used in the processing code.

5) **Processing code:** Interfacing Arduino and processing was learnt from-
<http://playground.arduino.cc/Interfacing/Processing>

Processing code- <https://github.com/Shubham-Shah/ITSP/blob/master/Processing/Processing.pde>

The incoming variable Total is split using divide and modulus function to get the x and y co-ordinates back. These are multiplied by a common factor and are then used to make lines.

In this way, when user moves his hand from one grid point to the other, a line is made between the two.
