

# KRUTAK

## Team: Unplugged

Arjun P S

Prachi Singh

Piyush Kumar Satpathy



# Smart Stylus



# PROBLEM STATEMENT

Nothing feels more natural than writing with your hand. The mouse can never be a replacement for a pen. Various Stylus' are available in the market but are all limited by the screen size of the device.

Design a pen that gets connected to your PC (or Hand-held device) such that everything written using the pen is made available on the connected device.

The handwritten content can be made available as a pastable drawing.

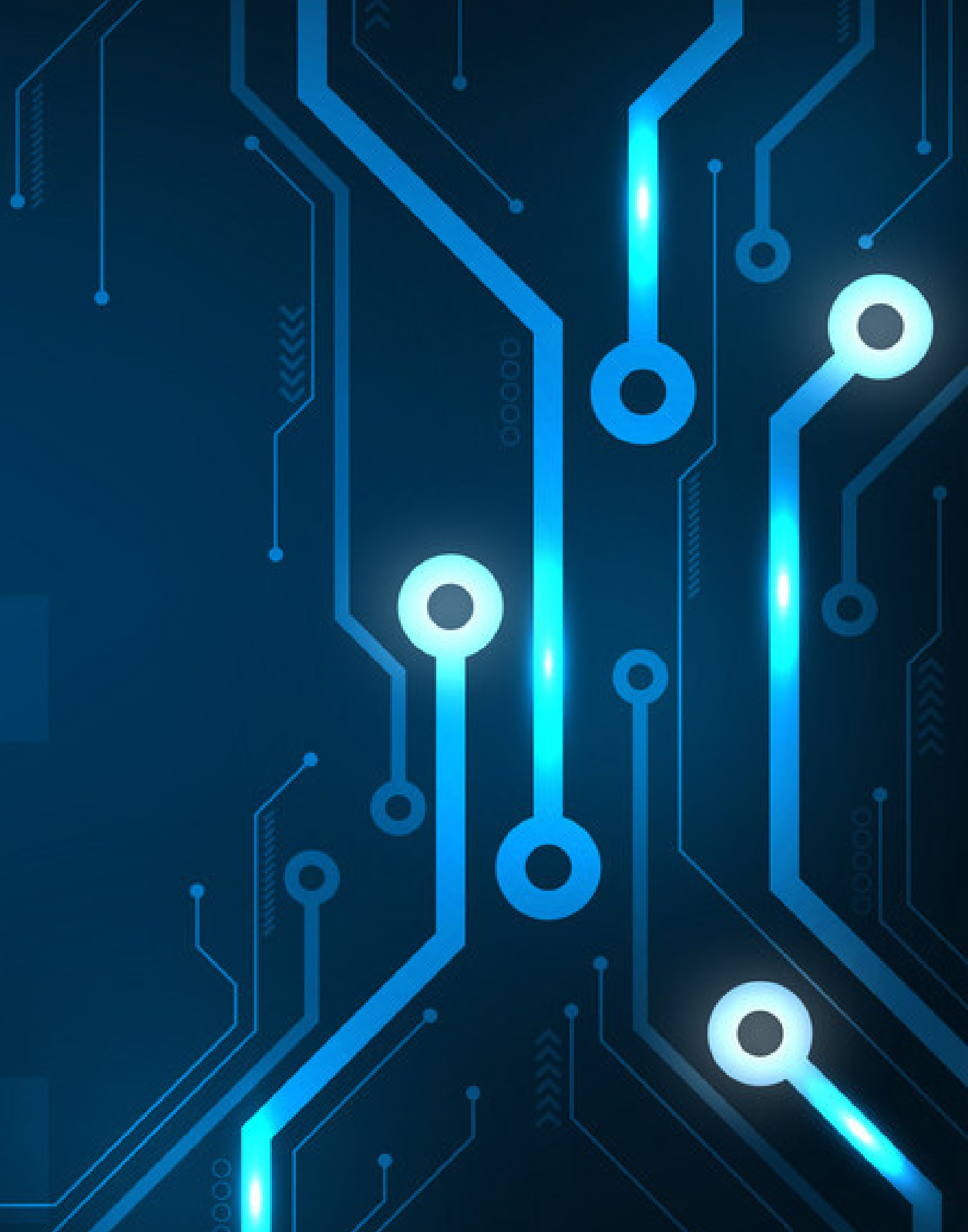
The pen should be usable on any surface, be it a tabletop or bulletin board.

The connection could be wired or wireless.



# BRIEF OVERVIEW

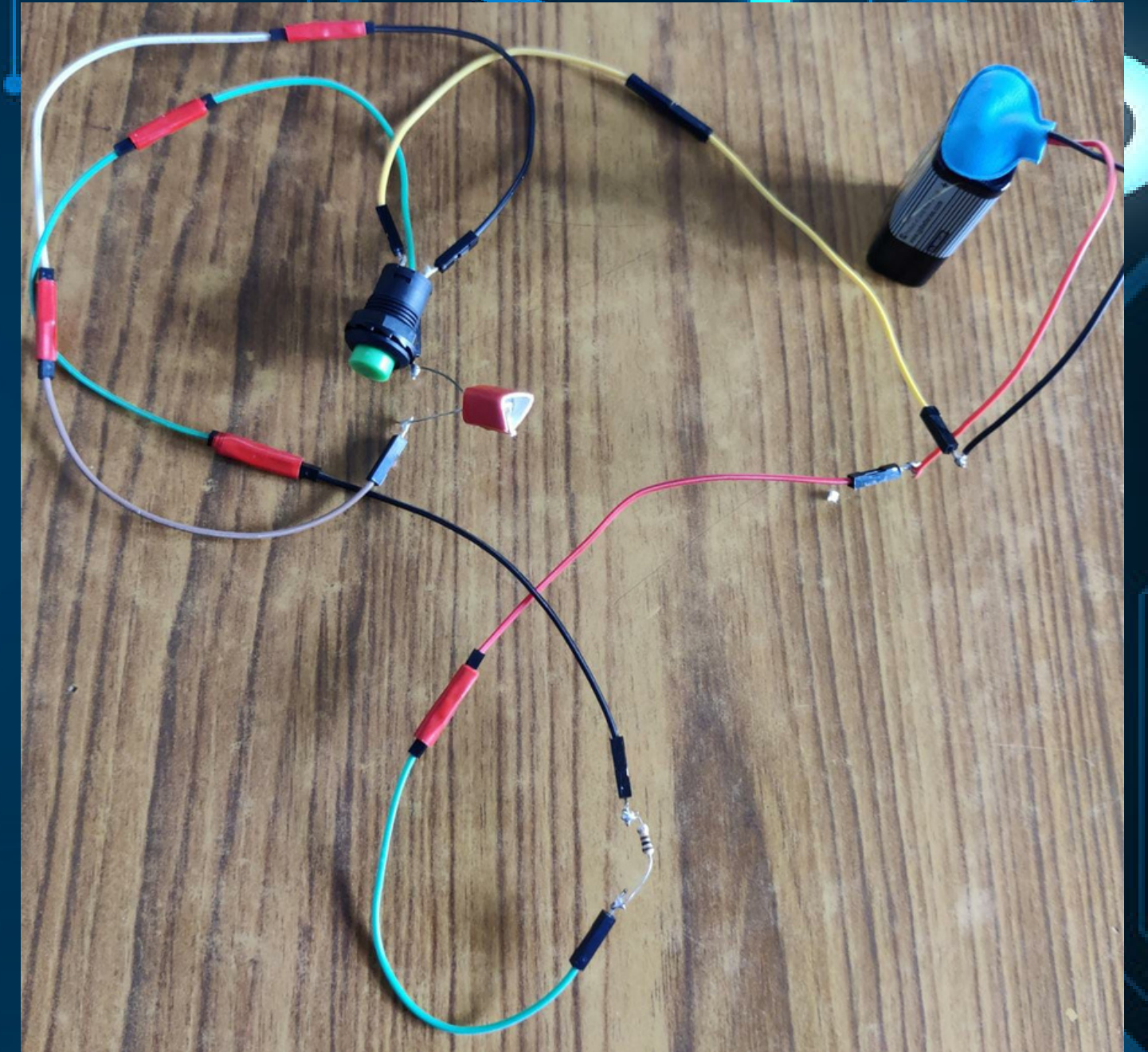
- In our project, we have used OpenCV in python to draw on any surface using a virtual pen.
- Smart Stylus provides us with a simpler and cost-effective version of a regular Stylus.





# SOLUTION

- Pen is made of led connected to a push button switch.
- User can use any camera, including the one on his mobile for using the stylus
- The camera detects the led (its colour and brightness) using HSV tracker.
- The python script used, takes in coordinates of led's center and draws line between present and past coordinate



# CODE

```
In [ ]: import cv2
import numpy as np
import time
import urllib.request
import os

wiper_thresh = 40000
kernel = np.ones((5,5),np.uint8)

canvas = None
clear = False

noiseth = 100

cx_prev = 0
cy_prev = 0

address = "address"
folder_path = "path"

i = 0

while True :

    image_resp = urllib.request.urlopen(address)
    image_np = np.array(bytearray(image_resp.read()), dtype = np.uint8)
    frame = cv2.imdecode(image_np, -1)
    (h, w, c) = frame.shape
    frame = cv2.resize(frame, (w//2, h//2))
    frame = cv2.flip(frame, 1)
    cv2.imshow("video", frame)
    frame = cv2.flip( frame, 1 )
    if canvas is None:
        canvas = np.zeros_like(frame)
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
```



```
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

lower_range = np.array([23,155,140])
upper_range = np.array([54,255,255])
mask = cv2.inRange(hsv, lower_range, upper_range)

#ret, mask = cv2.threshold(frame, 225, 255, cv2.THRESH_BINARY)

mask = cv2.erode(mask,kernel,iterations = 1)
mask = cv2.dilate(mask,kernel,iterations = 2)

contours, hierarchy = cv2.findContours(mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
if contours and cv2.contourArea(max(contours, key = cv2.contourArea)) > noiseth:

    c = max(contours, key = cv2.contourArea)
    area = cv2.contourArea(c)
    x2,y2,w,h = cv2.boundingRect(c)
    cx = x2 + w//2
    cy = y2 + h//2
    if cx_prev == 0 and cy_prev == 0:
        cx_prev, cy_prev = cx, cy

    else:
        canvas = cv2.line(canvas, (cx,cy),(cx_prev,cy_prev), (255,0,0), 4)
        cx_prev = cx
        cy_prev = cy
    if area > wiper_thresh:
        cv2.putText(canvas,'Clearing Canvas', (100,200),
            cv2.FONT_HERSHEY_SIMPLEX,2, (0,0,255), 5, cv2.LINE_AA)
        clear = True

    else:

        cx_prev,cy_prev =0,0

frame = cv2.add(frame,canvas)
stacked = np.hstack((canvas,frame))
cv2.imshow('PAGE',cv2.resize(stacked,None,fx=0.6,fy=0.6))
```



```
clear = True

else:

    cx_prev,cy_prev =0,0

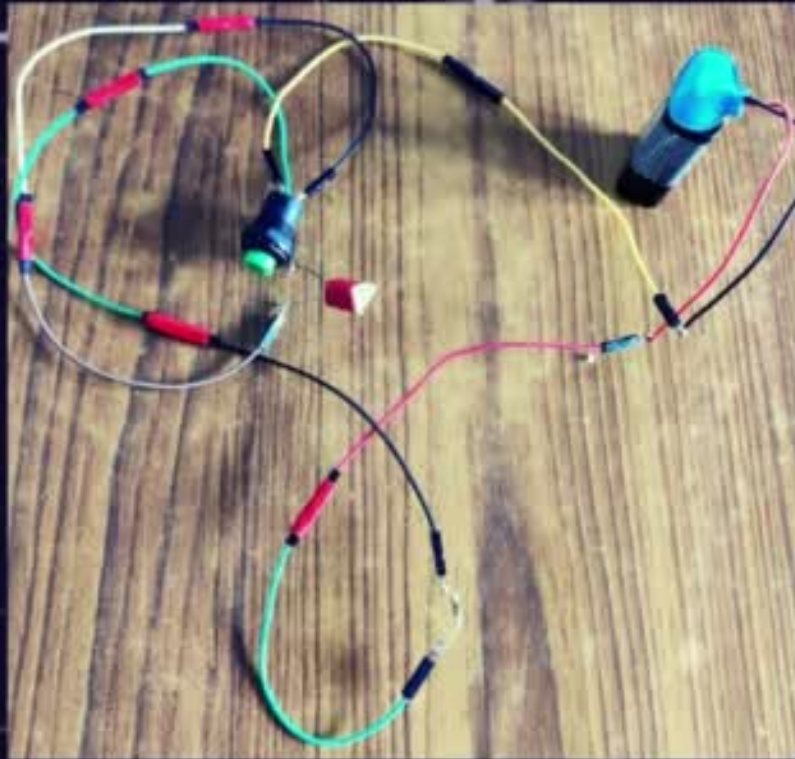
    frame = cv2.add(frame,canvas)
    stacked = np.hstack((canvas,frame))
    cv2.imshow('PAGE',cv2.resize(stacked,None,fx=0.6,fy=0.6))

    if clear == True:
        time.sleep(3)
        canvas = None
        clear = False
        x1, y1 = 0, 0
    k = cv2.waitKey(3) & 0xFF
    if k == 27:
        break
    if k == ord('c'):
        canvas = None

    if k == ord('s'):

        Image_save = cv2.resize(canvas, None, fx=0.6, fy=0.6)
        os.chdir(folder_path)
        cv2.imwrite('img_' + str(i) + '.jpg', Image_save)
        i += 1

    cv2.waitKey(1)
    cv2.destroyAllWindows()
```



# USER'S PERSPECTIVE

Adjusting the camera, the user will start the program and starts writing accordingly.

Our program tracks the movements of the stylus and returns the captured data

Once the writing is done, we press s to save the images into a directed folder.





# BILL OF MATERIAL

- Laptop / pc
- Stylus - Rs 30 (including price of battery)
- Camera (can be laptop's camera, mobile camera or any webcam)

# APPLICATION

- User can write on any surface (even in air)
- Very cheap (Rs 30 /-), so everyone can afford it.
- Higher life time and Easily replaceable materials materials, (battery and led)



# FURTHER DEVELOPMENTS

- Using ML algorithms to convert text in image to actual text format
- Using better tracking methods



# LIVE DEMONSTRATION



THANK YOU

