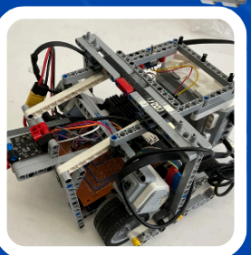
LINE_BUONARROTI

IL ROBOT:

Switch ON/OFF: Aziona il codice sul RaspberryPi tramite input digitale a resistenza di pull-up con pin GPIO

EV3: Riceve le potenze dal RapsberryPi e le invia ai motori.

RaspberryPi 4B: Il cervello del robot, che si occupa di interpretare i frame della telecamera e seguire la linea



Telecamera: Telecamera USB connessa al Rpi.

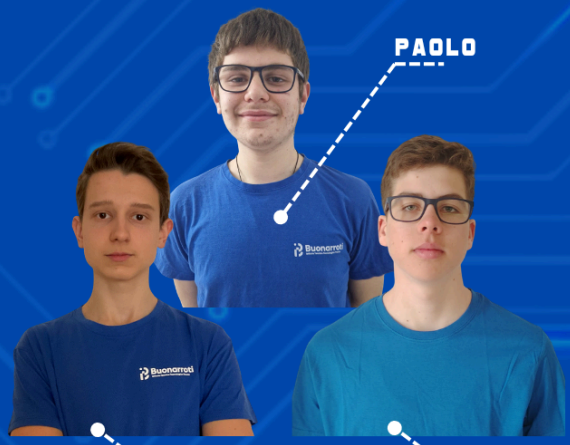
Pinza: Viene mossa da un motore lego

Sensori ToF: Sensori di distanza VL6180X ultraprecisi basati sulla velocità della luce

Motori: Motori da 160rpm e 40Ncm di coppia massima

Sensore Giroscopo: Giroscopio e accelerometro MPU6050 con più di 2000°/s

IL TEAM:



HIGHLIGHT DEL CODICE:

```
def get_blackbox(self, blackContours):
  blackCountours_len = len(blackContours)
  if blackCountours_len > 0:
    if blackCountours_len == 1:
      blackbox = cv2.minAreaRect(blackContours[0])
    else:
      candidates = []
      off_bottom = 0
      for con_num in range(blackCountours_len):
        blackbox = cv2.minAreaRect(blackContours[con_num])
        (x_min, y_min), (w_min, h_min), ang = blackbox
        box = cv2.boxPoints(blackbox)
        (x_box, y_box) = box[0]
        if y_box > self.height_cam - 2:
          off_bottom += 1
        candidates.append((y_box, con_num, x_min, y_min))
      candidates = sorted(candidates)
      if off_bottom > 1:
        candidates_off_bottom = []
        for con_num in range((blackCountours_len - off_bottom), blackCountours_len):
          (y_highest, con_highest, x_min, y_min) = candidates[con_num]
          total_distance = ((x_min - self.x_last)**2 + (y_min - self.y_last)**2)**0.5
          candidates_off_bottom.append((total_distance, con_highest))
        candidates_off_bottom = sorted(candidates_off_bottom)
        (_, con_highest) = candidates_off_bottom[0]
        blackbox = cv2.minAreaRect(blackContours[con_highest])
      else:
        (_, con_highest, x_min, y_min) = candidates[blackCountours_len - 1]
        blackbox = cv2.minAreaRect(blackContours[con_highest])
    return blackbox
  else:
    return None
```

Focus: Selezione Intelligente della Traiettoria
Invece di seguire "ciecamente" una macchia nera, il nostro algoritmo analizza la scena:
1. **Filtro Spaziale:** Identifica solo la linea nel raggio d'azione immediato.

2. **Memoria di Posizione:** Se la linea si divide, il robot calcola la distanza euclidea tra i nuovi percorsi e l'ultima posizione nota, scegliendo la continuità invece del caos.

PANORAMICA:

Oltre il semplice seguilinea: un veicolo autonomo che unisce meccanica di precisione e visione artificiale. Grazie all'integrazione tra sensori Laser ToF e telecamera, il robot interpreta il tracciato in tempo reale, superando ostacoli e incroci complessi con una fluidità di guida di livello superiore.

SOFTWARE:

Sviluppato in Python con libreria OpenCV, il sistema elabora i frame ad alta frequenza. L'architettura software filtra il rumore visivo e sceglie la traiettoria ottimale calcolando la distanza euclidea tra i contorni, garantendo decisioni rapide e coerenza nel tracking della linea.

ROBOCUP JUNIOR ACADEMY 2026
I.T. ARCHIMEDE – CATANIA
15-18 APRILE 2026

Buonarroti
Istituto Tecnico Tecnologico Trento

SCUOLA FUTURA