**Objective:**

For ECE 5725 Design with Embedded Operating System final project, we use a Raspberry Pi to control a robotic car to autonomously track a falling red balloon and follow it until popping the balloon. The camera can move up and down by tilt kit and the wheel could move around in the ground. Our car can also be remotely controlled by pygame interface in computer connecting with it in different modes.

**Introduction:**

We create a tracking robot with Raspberry Pi that can track a moving and falling red balloon and move to pop the balloon before it lands. We implemented the color detection to recognize the balloon by OpenCV. PID algorithm could calculate the suitable parameters to control the tilt kit of camera by hardware PWM precisely and stably and the movement of wheels by DC controller. Besides, python multiprocessor modules are implemented to decrease the latency which allows us to simultaneously capture video, process the images and calculate the parameters of PID control and send the command to the motors.

**Design and Testing:**

We only use one pi camera to detect objects and measure the distance between the object and the car. Our solution is to move the car according to the location of the balloon in the images and the criterion is to keep the balloon in the center of the images, which means the direction of moving car is in the line with that of the balloon. Additionally, the area of the balloon in the image can reflect the value of the distance between them. So it requires the accuracy of the recognition to the balloon.

In order to accomplish the project better, the best approach was to divide it into various smaller parts individually and combine them at the end. The main task is hardware and software part, which are tackled different task as following introduction.

**Hardware:**

3D printer:

DC controller:

**Software:**

1. **Object Recognition Algorithm:**

Our object recognition algorithm could detect the red objects videoed by the Pi camera and calculate the center position. The usage of the OpenCV library makes the algorithm easy to implement. First of all, we install the openCV library into our Raspberry Pi. There are some ways to install the openCV library. The easiest and direct method is to run the command “*sudo apt-get install libopencv-dev python-opencv*” in the command prompt, which can help us to save several hours to compile the openCV. The version of our installed openCV is ***这个需要去查一下。*** After installing the openCV successfully, we need to assemble the Pi camera onto the R-Pi and set it up in configuration of R-Pi. As for the design of the recognition algorithm, it can be broken up into following parts.

1. Create a video capture object of openCV which can take picture at real time by running “cv2.VideoCapture(0)” and when using “cap.read()”, it will return 640\*480\*3 picture array, which is the default resolution and can fit our requirement of recognition. Three channels is (R, G, B) color space.
2. Run “cv2.cvtColor(frame, cv2.COLOR\_RGB2HSV)” can convert the image from RGB color space to HSV (hue, Saturation, Value) color space. We can benefit from this conversion because we need to threshold the HSV image to get the binary image which only keep the red balloon part by setting the low\_red and upper\_red threshold and it is convenient and easy for us to adjust the threshold in HSV color space.
3. We set the lower\_red is (156, 100, 40) and the upper\_red is (180, 255, 255). Use the syntax of “maks = cv2.inRange(hsv, low\_red, upper\_red)” to do threshold. It will make the pixel values that are less than lower\_red and more than upper\_red equal to zero and the pixel values that are in between them equal to 255.
4. We also use the techniques of computer vision to process the binary images including do “cv2.morphologyEx” of open and close to filter the background noise and make the boundary of binary image smooth and clear.
5. Run “cv2.findContours(mask,cv2.RETR\_EXTERNAL,cv2.CHAIN\_APPROX\_NONE)” to do the most important thing which is extract the boundary of the object. If it detects the boundary, it will return a list that contains several calculated contours although most of them are zero. If it does not, it will return a empty list.
6. We need to pick up the longest contour from returned contour list. The method we used is to calculate the area of every contour and find the maximum value that is what we want. Then, calculate the spatial moment that can compute the center position of the object and the area of the object. s
7. **PID control Algorithm:**

The size of image is 640\*480 and thus the center of the image should be (320, 240). The object recognition algorithm should compute the center position of the object. Compute their difference and use it as feedback to control the servo and make the camera stare at the object, thereby achieving the camera tracking. Besides, the motors of wheel also need to be controlled to move suitably to track the balloon.

1. **Multi-processors Algorithm:**
2. **Prediction Algorithm:**

**Issue:**

1. We tried to install the C++ OpenCV library which takes ore than three hours to compile it and after it successfully install, it sti