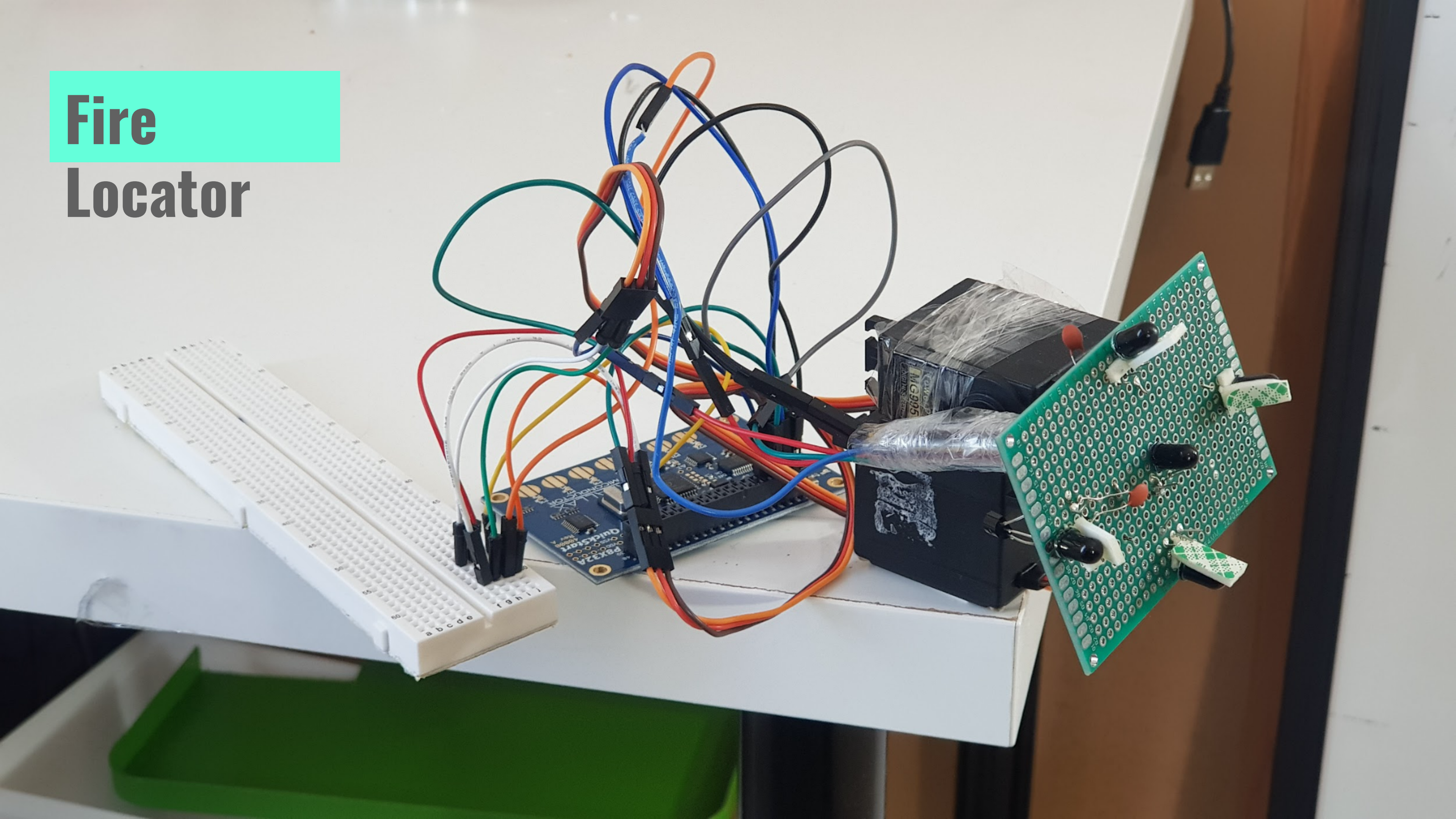
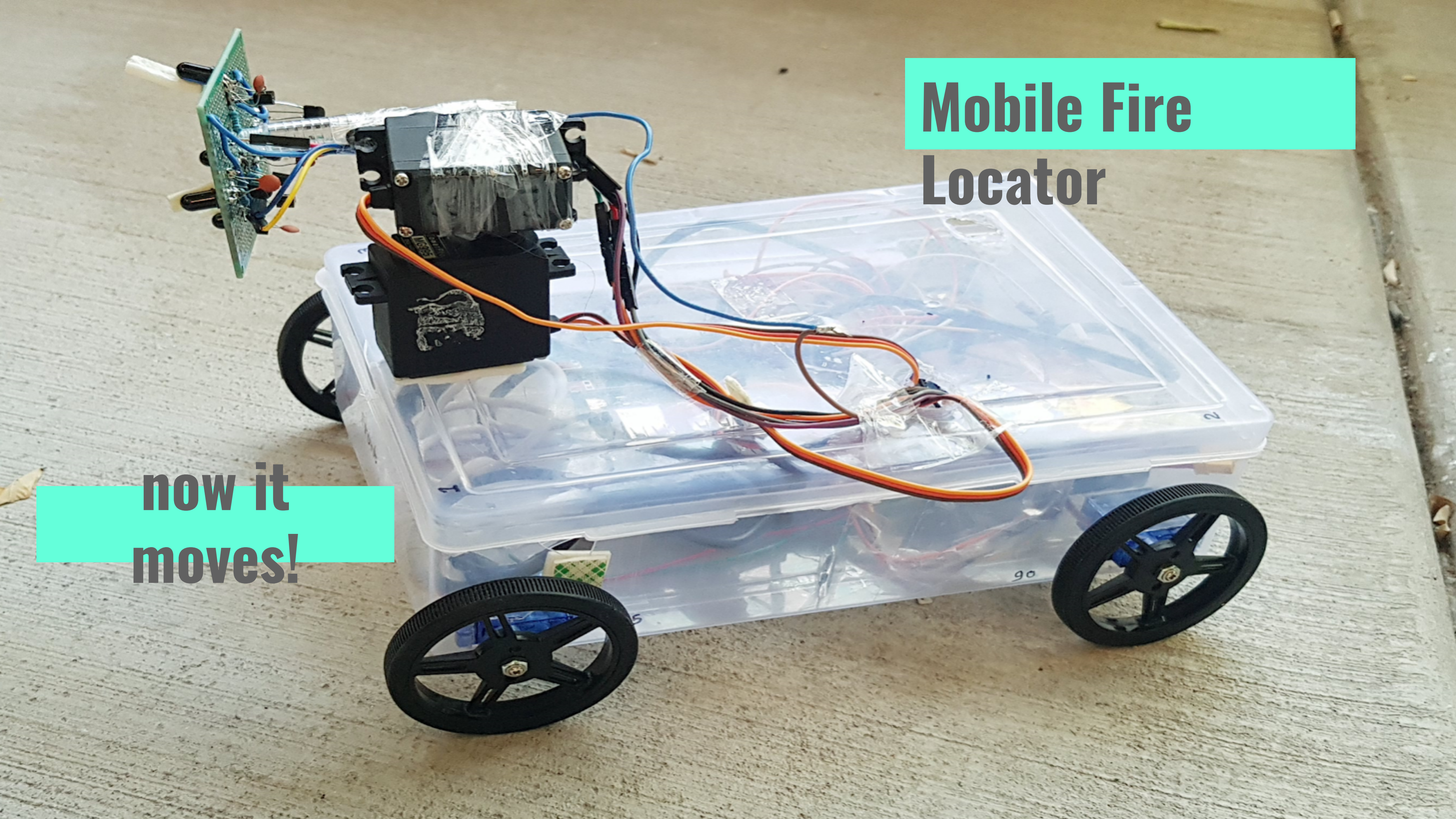
Mobile Fire Locator

Gaurav Nawale, Sreeja Vangapelli, Diego Pozo

Fire Locator





Mobile Fire Locator

now it moves!

Modules

Old

- Sensing
  - IR
  - Light
  - Temperature
- Processing
  - Filtering
  - Locating
- Moving
  - Orientation
  - Position

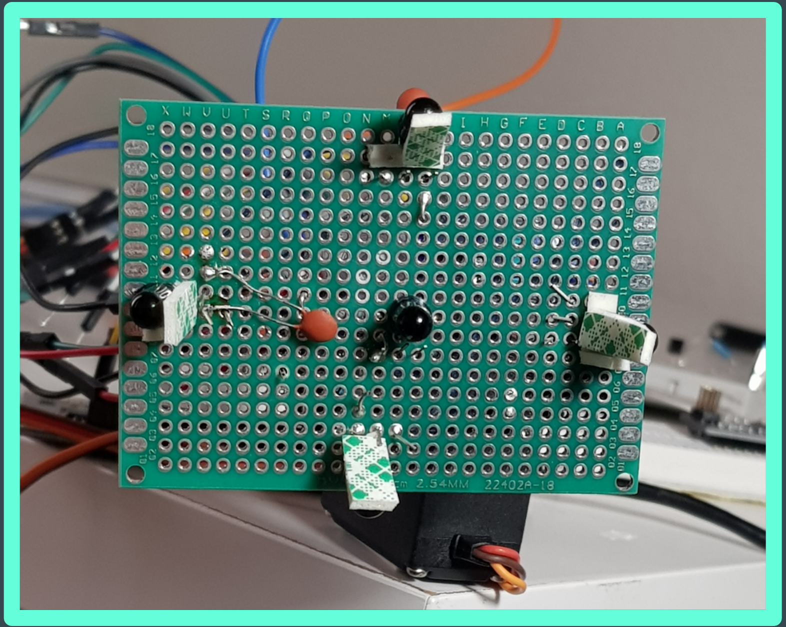
Modules

New

- Sensing (RasPi)
  - IR
  - Adapt to Raspberry Pi
- Processing
  - Filtering
  - Locating
- Moving (Arduino)
  - Orientation (sensor head)
  - Position (cart)
- Communication
  - UART

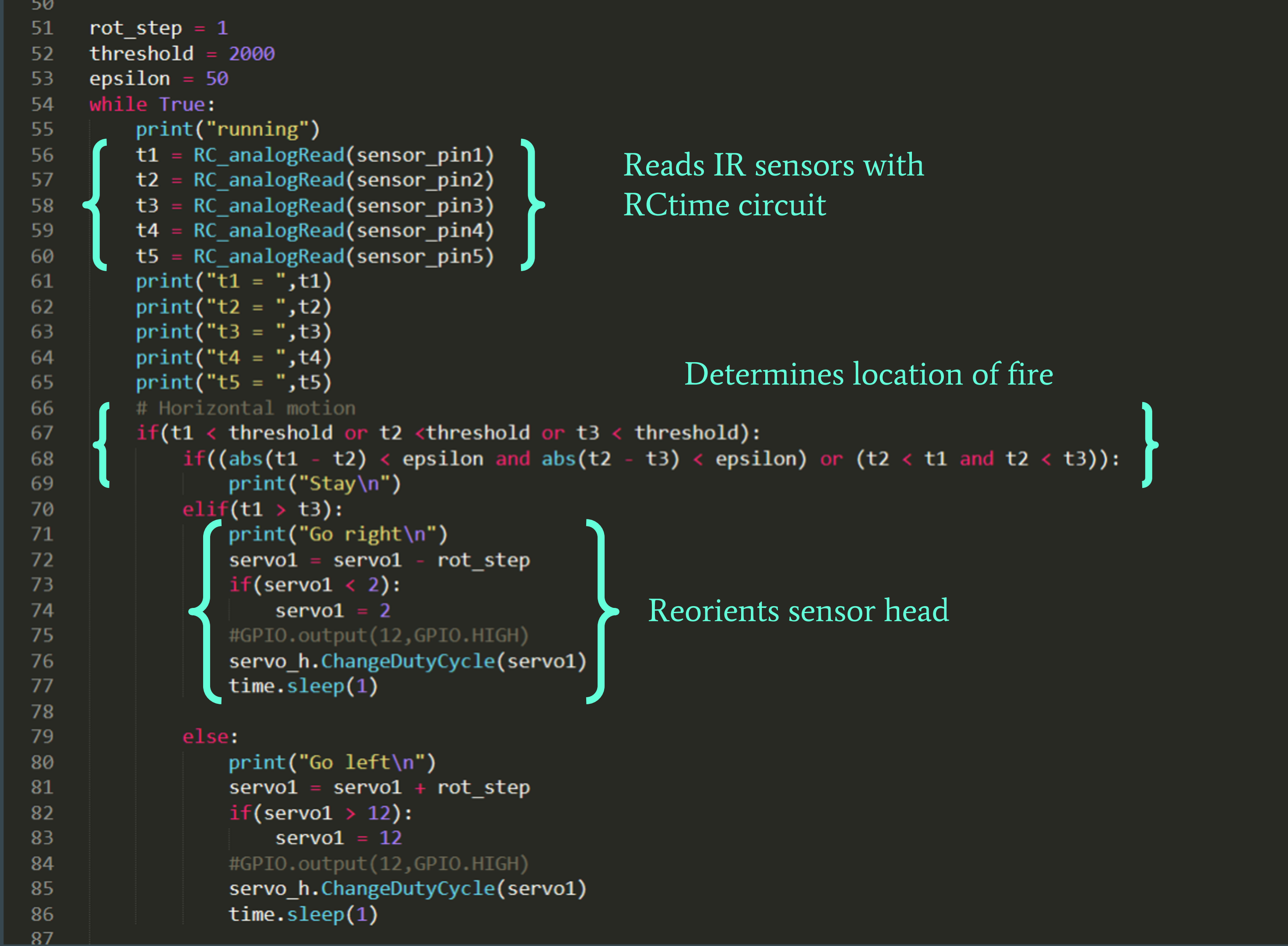
Sensor head

- Raspberry Pi
  - Turns independently from body
  - Two different axes
  - Senses intensity / distance from fire



Code

Sensor head



```python
rot_step = 1
threshold = 2000
epsilon = 50
while True:
    print("running")
    t1 = RC_analogRead(sensor_pin1)
    t2 = RC_analogRead(sensor_pin2)
    t3 = RC_analogRead(sensor_pin3)
    t4 = RC_analogRead(sensor_pin4)
    t5 = RC_analogRead(sensor_pin5)
    print("t1 = ",t1)
    print("t2 = ",t2)
    print("t3 = ",t3)
    print("t4 = ",t4)
    print("t5 = ",t5)
    # Horizontal motion
    if(t1 < threshold or t2 <threshold or t3 < threshold):
        if((abs(t1 - t2) < epsilon and abs(t2 - t3) < epsilon) or (t2 < t1 and t2 < t3)):
            print("Stay\n")
        elif(t1 > t3):
            print("Go right\n")
            servo1 = servo1 - rot_step
            if(servo1 < 2):
                servo1 = 2
            #GPIO.output(12,GPIO.HIGH)
            servo_h.ChangeDutyCycle(servo1)
            time.sleep(1)

        else:
            print("Go left\n")
            servo1 = servo1 + rot_step
            if(servo1 > 12):
                servo1 = 12
            #GPIO.output(12,GPIO.HIGH)
            servo_h.ChangeDutyCycle(servo1)
            time.sleep(1)
```

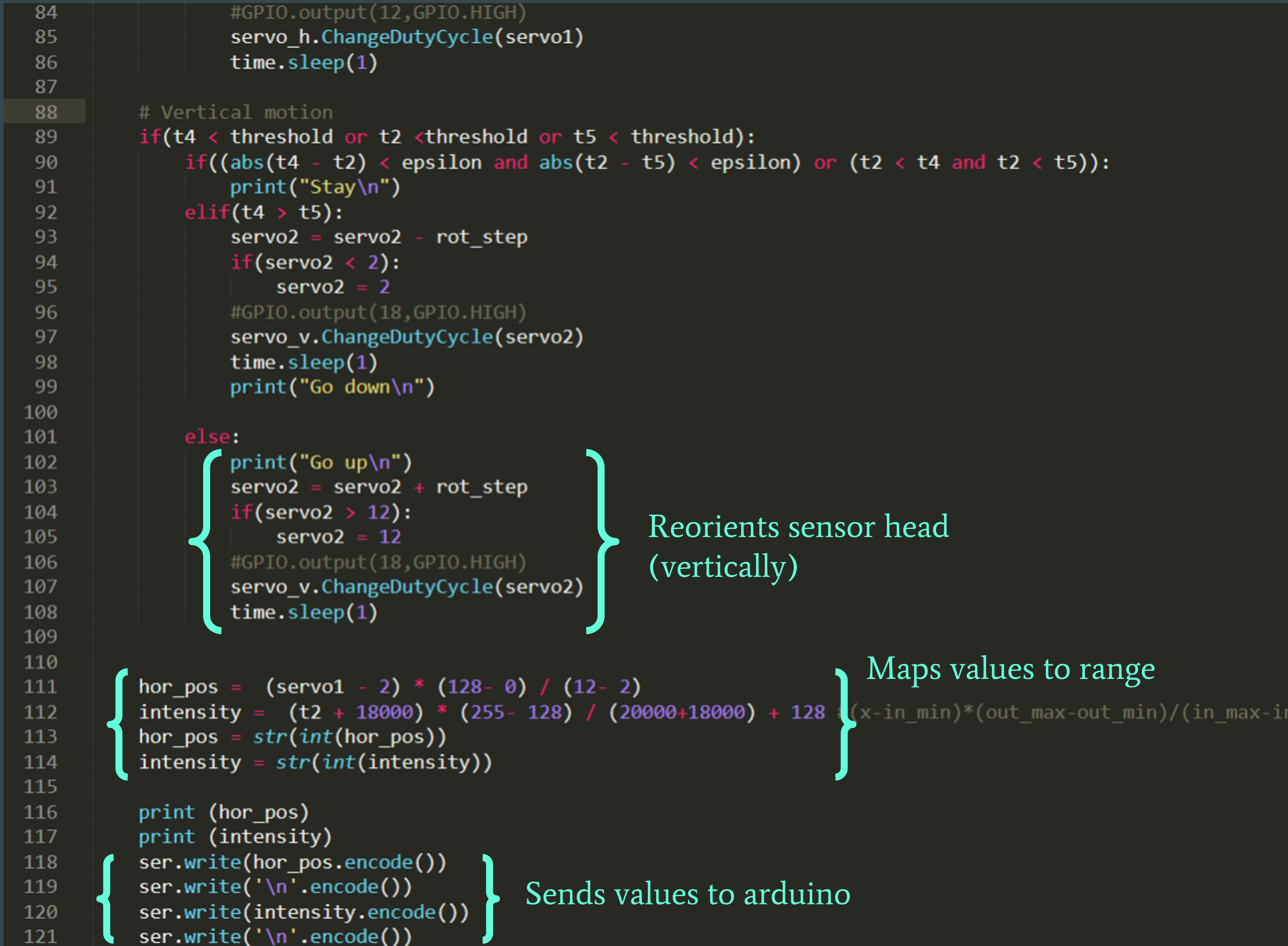
Reads IR sensors with RCtime circuit

Determines location of fire

Reorients sensor head

Code

Sensor head



```python
            #GPIO.output(12,GPIO.HIGH)
            servo_h.ChangeDutyCycle(servo1)
            time.sleep(1)

    # Vertical motion
    if(t4 < threshold or t2 <threshold or t5 < threshold):
        if((abs(t4 - t2) < epsilon and abs(t2 - t5) < epsilon) or (t2 < t4 and t2 < t5)):
            print("Stay\n")
        elif(t4 > t5):
            servo2 = servo2 - rot_step
            if(servo2 < 2):
                servo2 = 2
            #GPIO.output(18,GPIO.HIGH)
            servo_v.ChangeDutyCycle(servo2)
            time.sleep(1)
            print("Go down\n")

        else:
            print("Go up\n")
            servo2 = servo2 + rot_step
            if(servo2 > 12):
                servo2 = 12
            #GPIO.output(18,GPIO.HIGH)
            servo_v.ChangeDutyCycle(servo2)
            time.sleep(1)

    hor_pos =  (servo1 - 2) * (128- 0) / (12- 2)
    intensity =  (t2 + 18000) * (255- 128) / (20000+18000) + 128 #(x-in_min)*(out_max-out_min)/(in_max-i
    hor_pos = str(int(hor_pos))
    intensity = str(int(intensity))

    print (hor_pos)
    print (intensity)
    ser.write(hor_pos.encode())
    ser.write('\n'.encode())
    ser.write(intensity.encode())
    ser.write('\n'.encode())
```

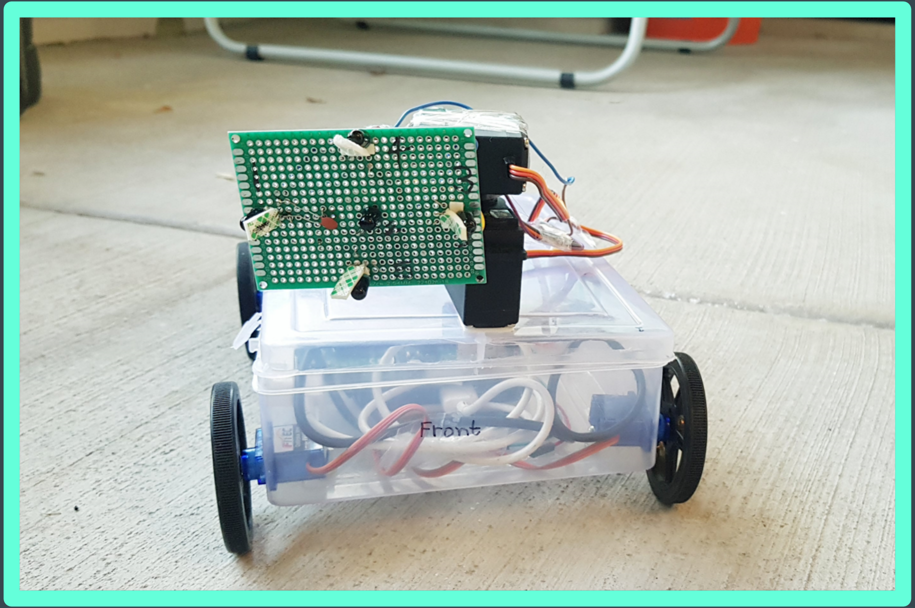
Reorients sensor head (vertically)

Maps values to range

Sends values to arduino

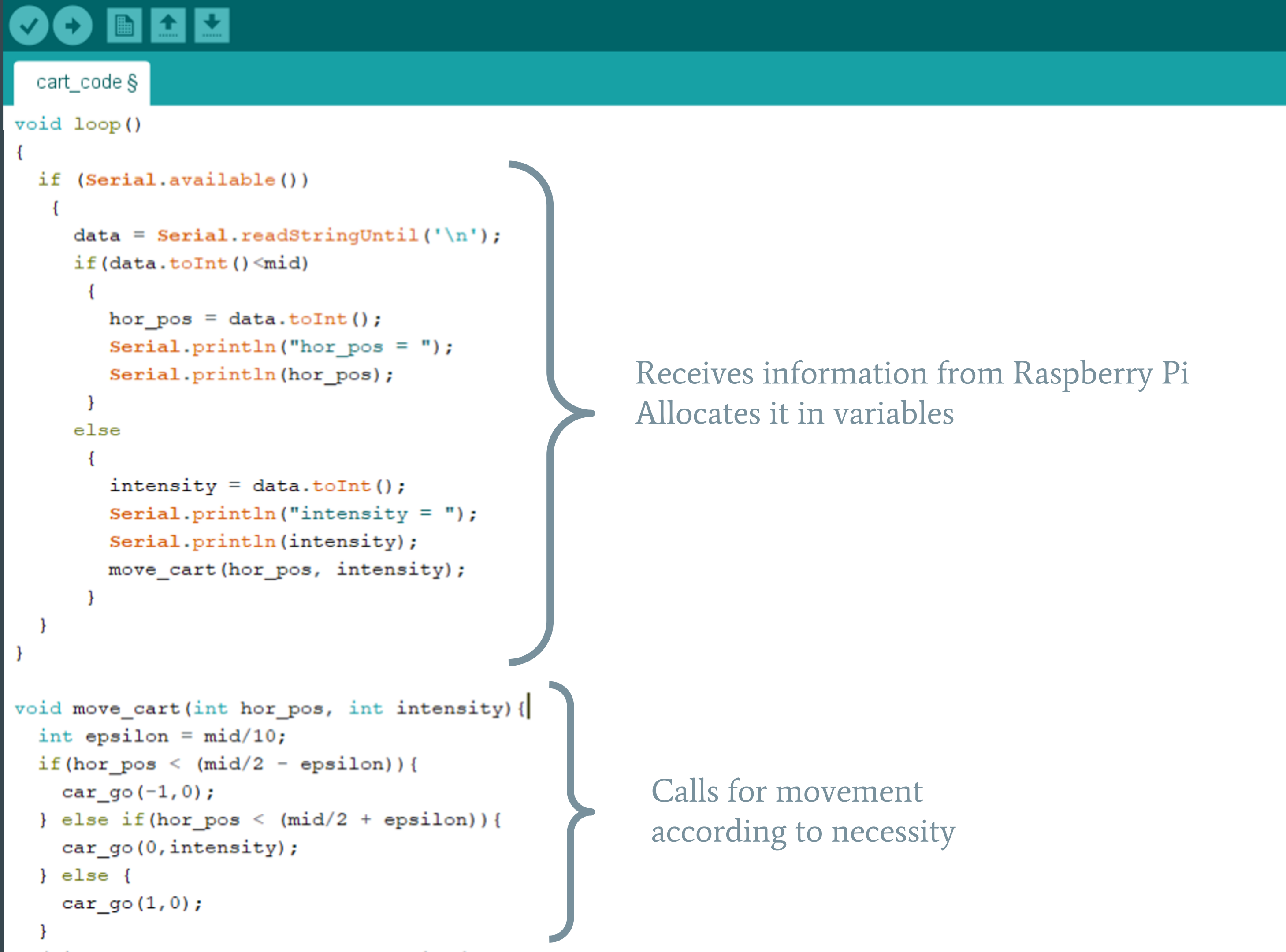
Mobile Cart

- Arduino
  - Four different states
    - Movement
      - Turn right
      - Turn left
      - Approach
    - Actuation
      - Extinguisher (buzzer)



Processin

Mobile cart



```
void loop()
{
  if (Serial.available())
   {
     data = Serial.readStringUntil('\n');
     if(data.toInt()<mid)
      {
        hor_pos = data.toInt();
        Serial.println("hor_pos = ");
        Serial.println(hor_pos);
      }
     else
      {
        intensity = data.toInt();
        Serial.println("intensity = ");
        Serial.println(intensity);
        move_cart(hor_pos, intensity);
      }
  }
}

void move_cart(int hor_pos, int intensity){
  int epsilon = mid/10;
  if(hor_pos < (mid/2 - epsilon)){
    car_go(-1,0);
  } else if(hor_pos < (mid/2 + epsilon)){
    car_go(0,intensity);
  } else {
    car_go(1,0);
  }
```

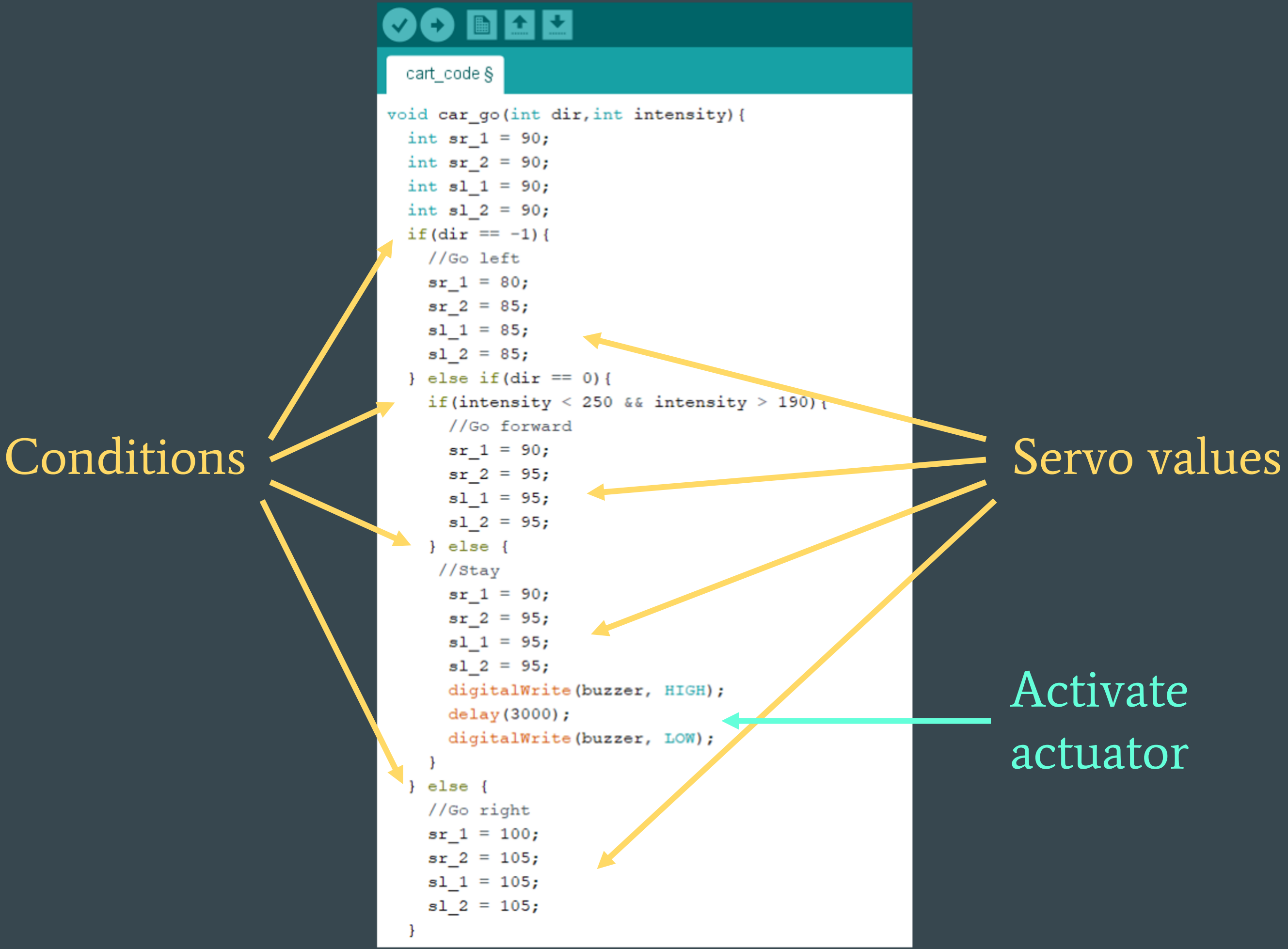
Receives information from Raspberry Pi
Allocates it in variables

Calls for movement
according to necessity

Processing

Mobile cart

- States

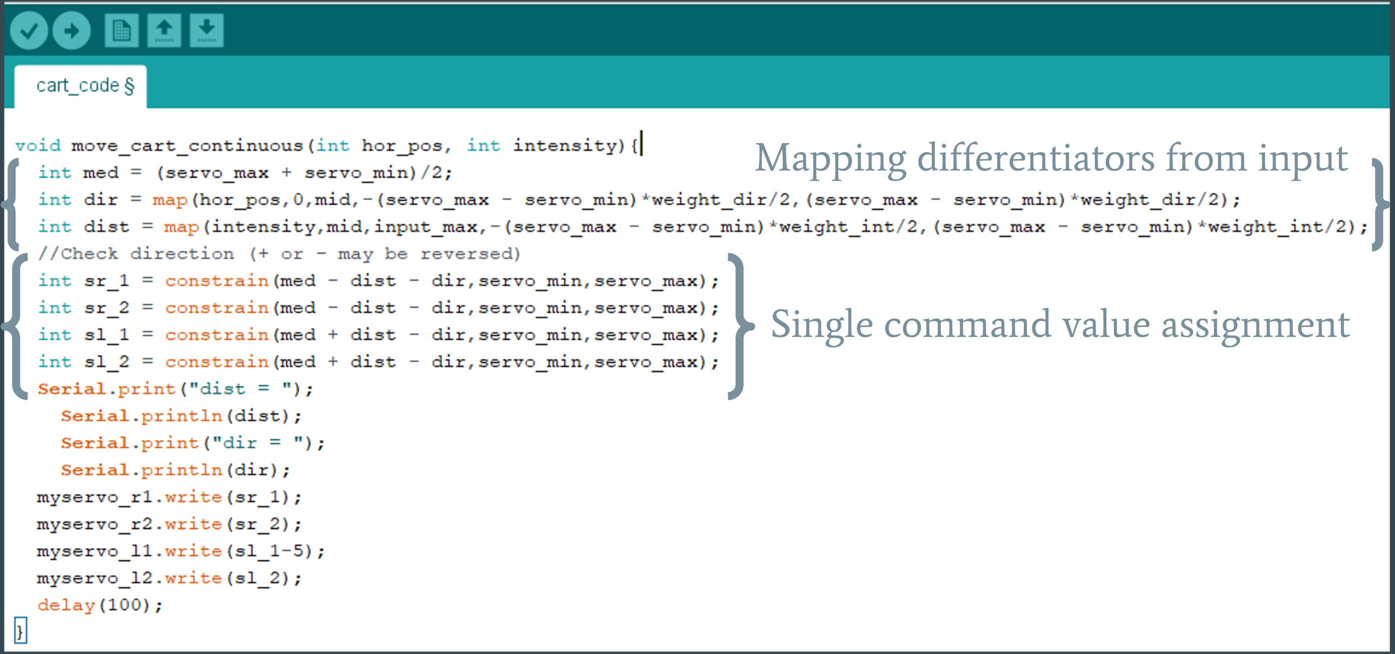


```
void car_go(int dir,int intensity){
  int sr_1 = 90;
  int sr_2 = 90;
  int sl_1 = 90;
  int sl_2 = 90;
  if(dir == -1){
    //Go left
    sr_1 = 80;
    sr_2 = 85;
    sl_1 = 85;
    sl_2 = 85;
  } else if(dir == 0){
    if(intensity < 250 && intensity > 190){
      //Go forward
      sr_1 = 90;
      sr_2 = 95;
      sl_1 = 95;
      sl_2 = 95;
    } else {
     //Stay
      sr_1 = 90;
      sr_2 = 95;
      sl_1 = 95;
      sl_2 = 95;
      digitalWrite(buzzer, HIGH);
      delay(3000);
      digitalWrite(buzzer, LOW);
    }
  } else {
    //Go right
    sr_1 = 100;
    sr_2 = 105;
    sl_1 = 105;
    sl_2 = 105;
  }
```

Processing

Mobile cart

- Continuous



```
void move_cart_continuous(int hor_pos, int intensity){
  int med = (servo_max + servo_min)/2;
  int dir = map(hor_pos,0,mid,-(servo_max - servo_min)*weight_dir/2,(servo_max - servo_min)*weight_dir/2);
  int dist = map(intensity,mid,input_max,-(servo_max - servo_min)*weight_int/2,(servo_max - servo_min)*weight_int/2);
  //Check direction (+ or - may be reversed)
  int sr_1 = constrain(med - dist - dir,servo_min,servo_max);
  int sr_2 = constrain(med - dist - dir,servo_min,servo_max);
  int sl_1 = constrain(med + dist - dir,servo_min,servo_max);
  int sl_2 = constrain(med + dist - dir,servo_min,servo_max);
  Serial.print("dist = ");
    Serial.println(dist);
    Serial.print("dir = ");
    Serial.println(dir);
  myservo_r1.write(sr_1);
  myservo_r2.write(sr_2);
  myservo_l1.write(sl_1-5);
  myservo_l2.write(sl_2);
  delay(100);
}
```

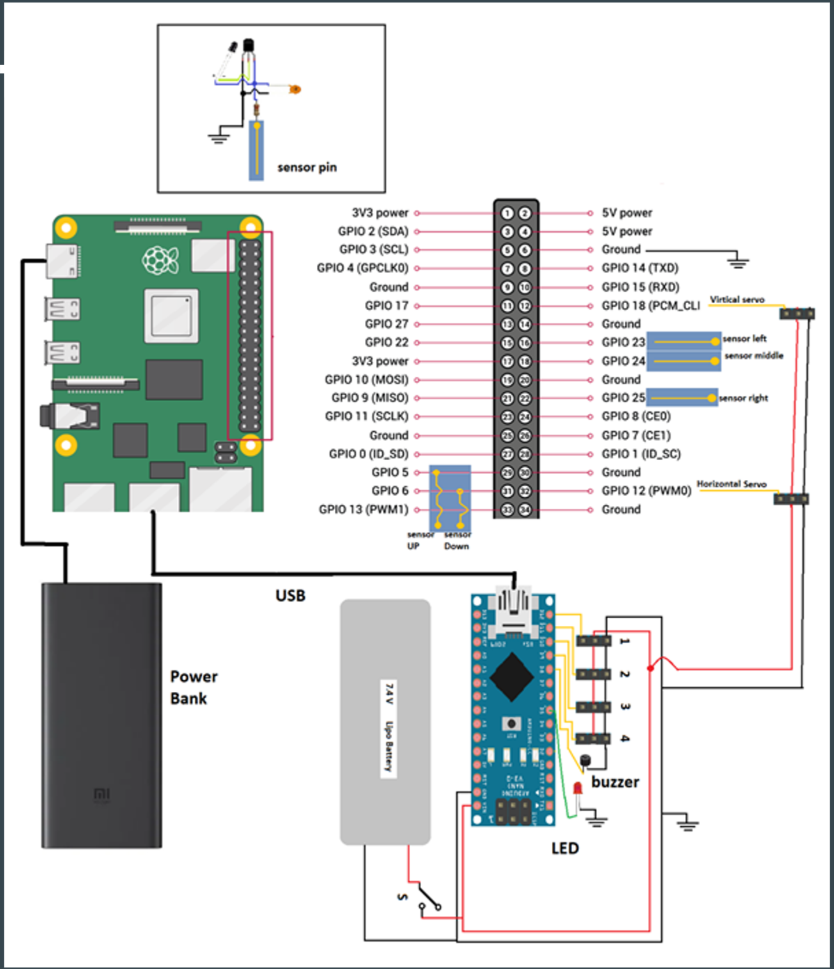
Mapping differentiators from input

Single command value assignment

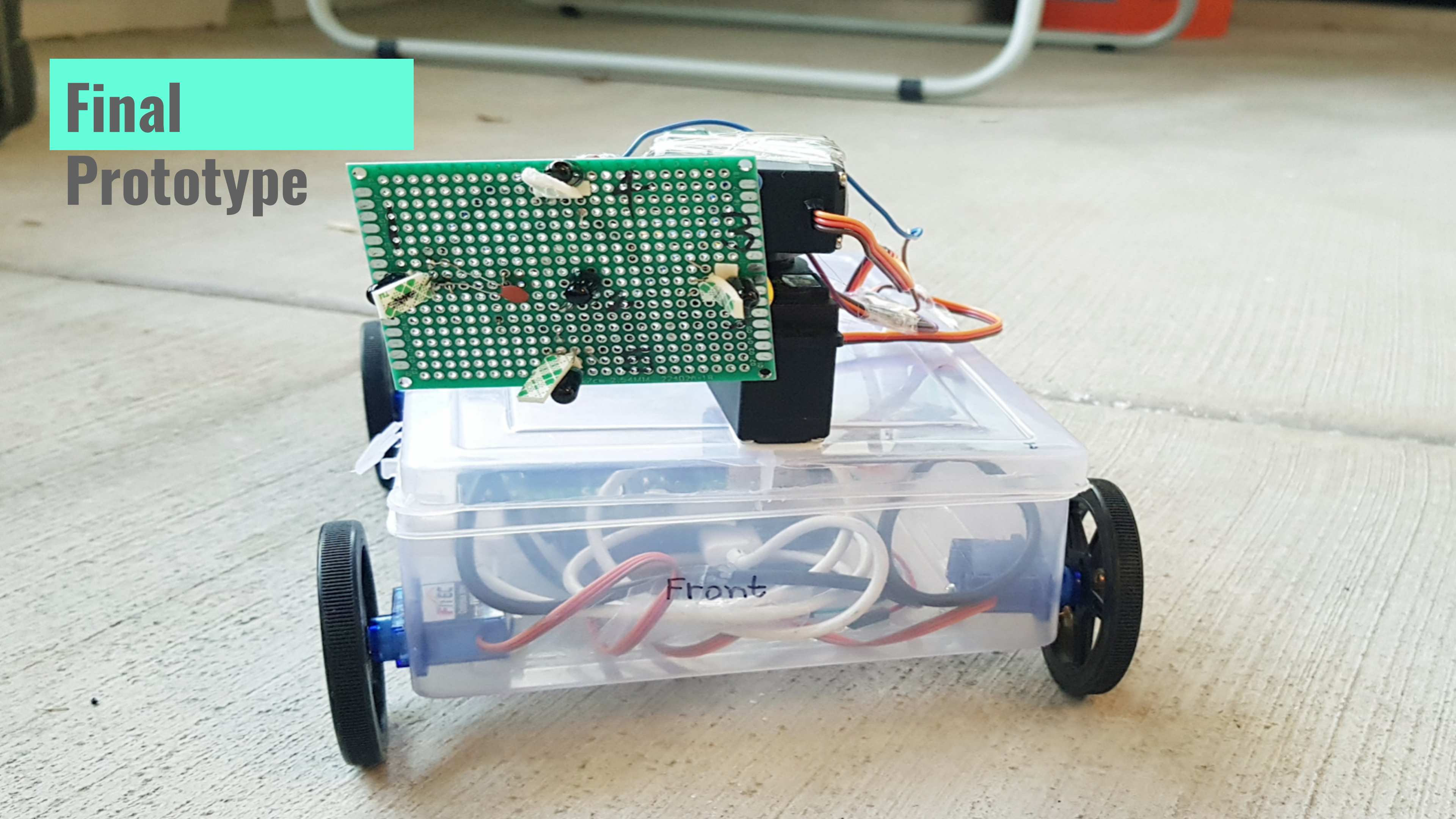
Bill of materials

Manufacturer	Part NAme	Part Type	Quantity	Price
Raspberry Pi foundation	Single board computer	Raspberry Pi 4	1	$40.00
Arduino	Microcontroller	Arduino nano	1	$21.00
Adafruit	Continuous servo motors	FS90R	4	$27.99
TowerPro	Standard metal Servo Motor	MG-995	2	$11.59
Uxcell	Infrared Sensor	IR Receiver photodiode	5	$1.50
-	PCB Board	-	1	$2.00
			Total	$104.08

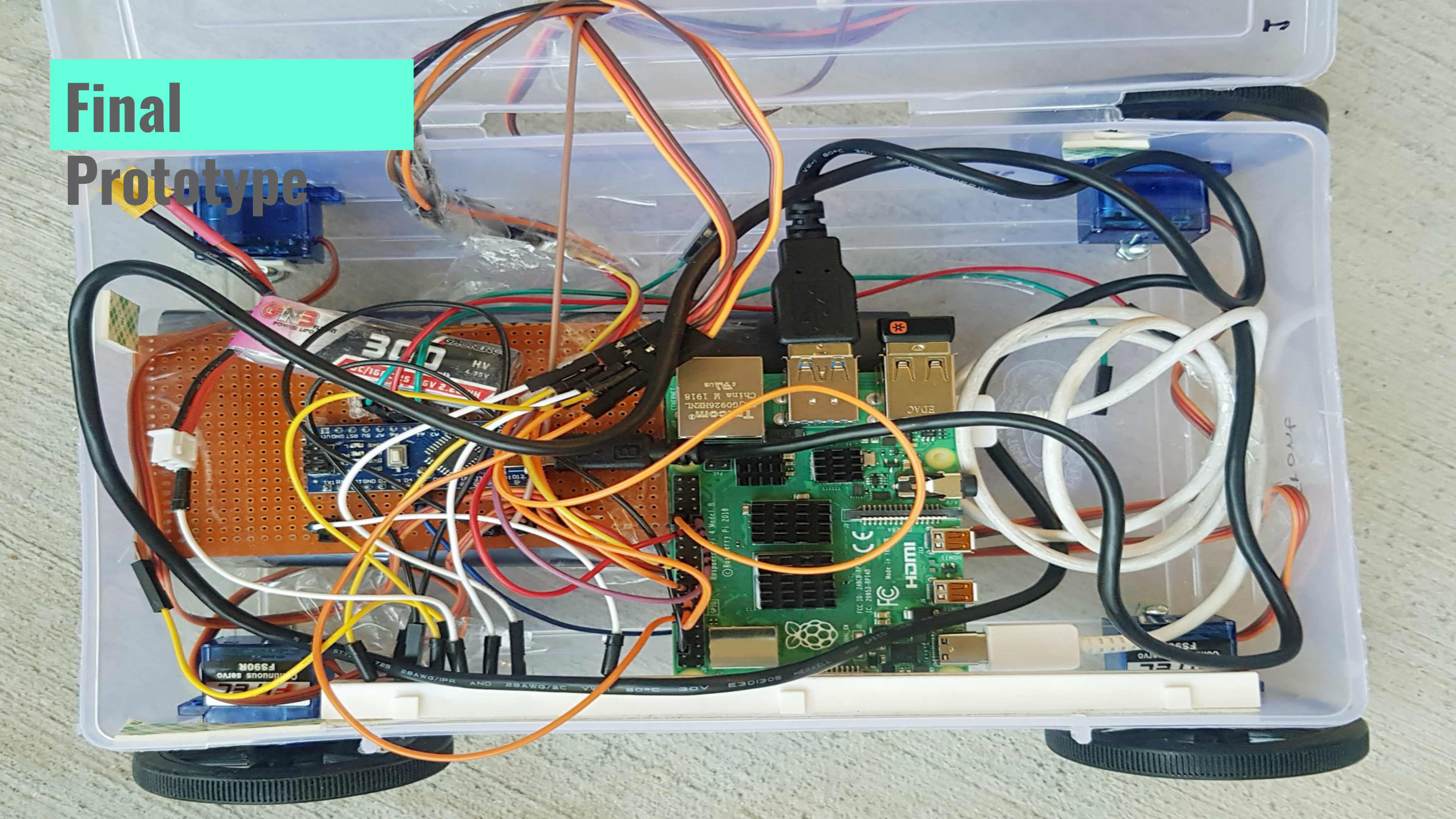
Electric circui



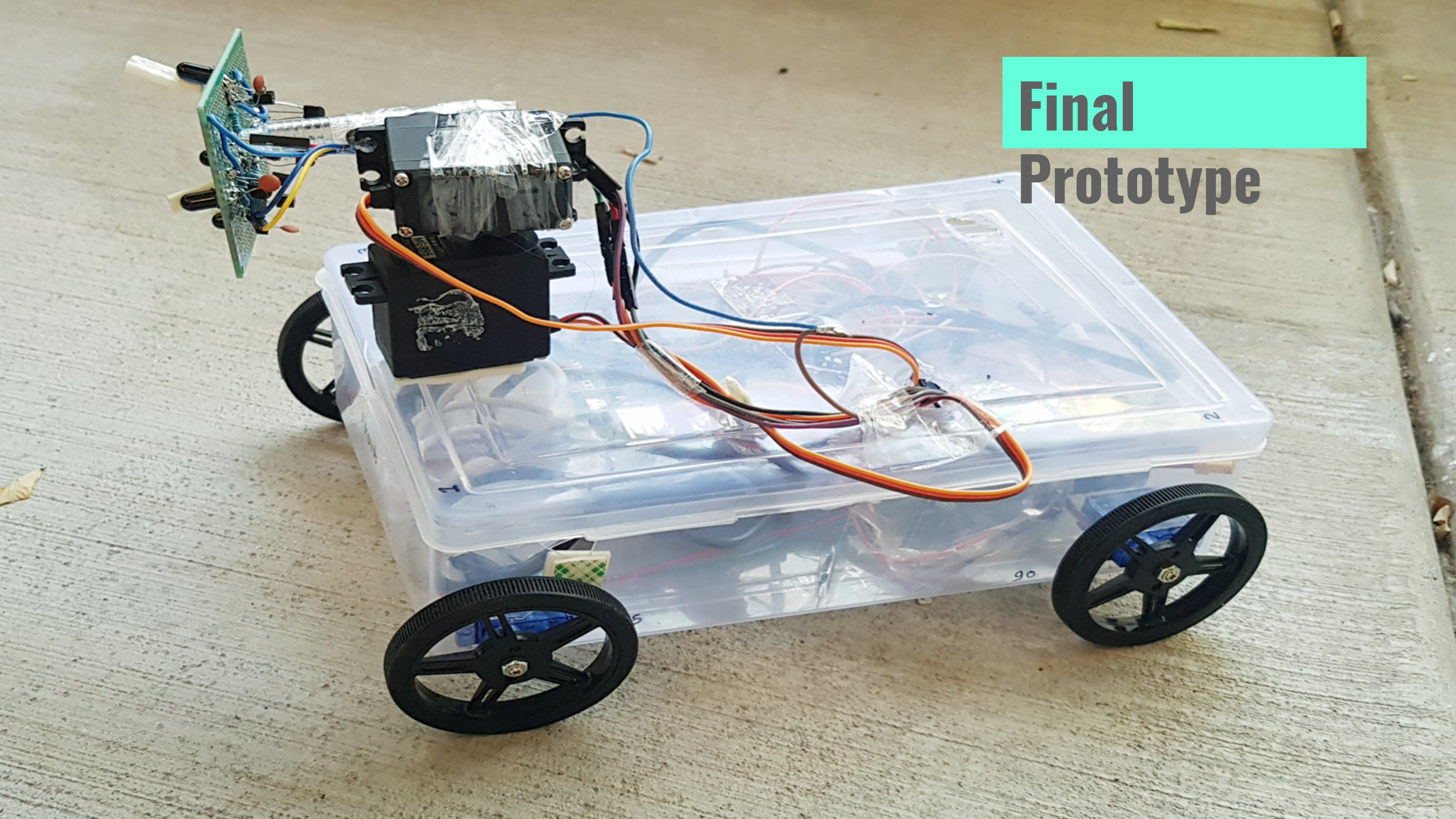
Final Prototype



Final Prototype



Final Prototype



Final product