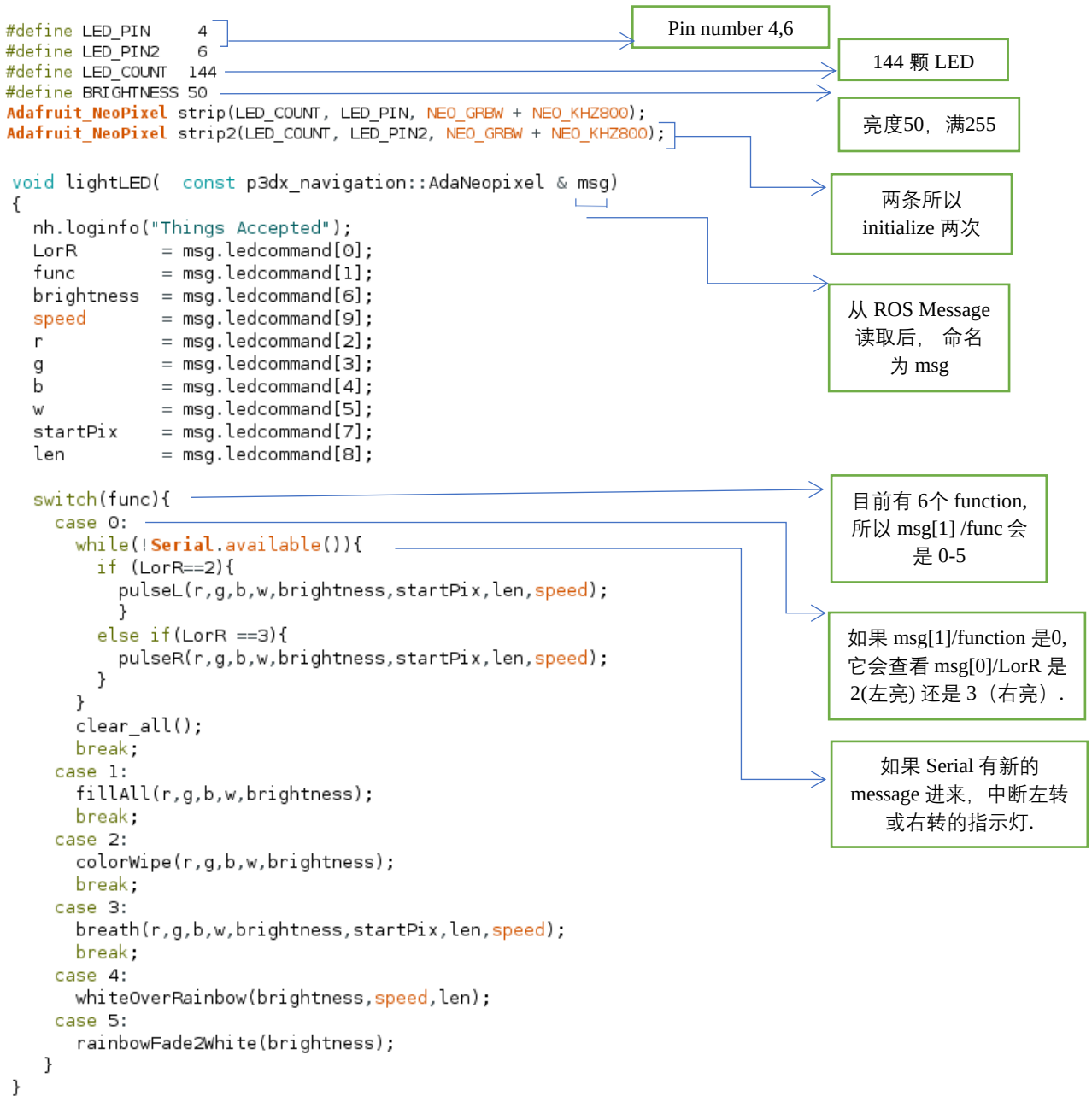

#define LED_PIN 4
#define LED_PIN2 6
#define LED_COUNT 144
#define BRIGHTNESS 50
Adafruit_NeoPixel strip(LED_COUNT, LED_PIN, NEO_GRBW + NEO_KHZ800);
Adafruit_NeoPixel strip2(LED_COUNT, LED_PIN2, NEO_GRBW + NEO_KHZ800);
void lightLED(const p3dx_navigation::AdaNeopixel & msg)
{
nh.loginfo("Things Accepted");
LorR = msg.ledcommand[0];
func = msg.ledcommand[1];
brightness = msg.ledcommand[6];
speed = msg.ledcommand[9];
r = msg.ledcommand[2];
g = msg.ledcommand[3];
b = msg.ledcommand[4];
w = msg.ledcommand[5];
startPix = msg.ledcommand[7];
len = msg.ledcommand[8];
switch(func){
case 0:
while(!Serial.available()){
if (LorR==2){
pulseL(r,g,b,w,brightness,startPix,len,speed);
}
else if(LorR ==3){
pulseR(r,g,b,w,brightness,startPix,len,speed);
}
}
clear_all();
break;
case 1:
fillAll(r,g,b,w,brightness);
break;
case 2:
colorWipe(r,g,b,w,brightness);
break;
case 3:
breath(r,g,b,w,brightness,startPix,len,speed);
break;
case 4:
whiteOverRainbow(brightness,speed,len);
case 5:
rainbowFade2White(brightness);
}
}
Pin number 4,6
144 颗 LED
亮度50，满255
两条所以 initialize 两次
从 ROS Message 读取后， 命名为 msg
目前有 6个 function, 所以 msg[1] /func 会是 0-5
如果 msg[1]/function 是0, 它会查看 msg[0]/LorR 是 2(左亮) 还是 3（右亮）.
如果 Serial 有新的 message 进来，中断左转或右转的指示灯.

```
ros::Subscriber<p3dx_navigation::AdaNeopixel> sub("/Neopixel" , lightLED);

void setup()
{
  //AdafruitNeopixel
  #if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)
      clock_prescale_set(clock_div_1);
  #endif
  strip.begin();     // INITIALIZE NeoPixel strip object (REQUIRED)
  strip2.begin();
  strip.show();      // Turn OFF all pixels ASAP
  strip2.show();
  nh.initNode();
  nh.subscribe(sub);
  delay(100);

}

void loop()   {

  nh.spinOnce();
  delay(1);

}


void whiteOverRainbow(int brightness, int speed, int whiteLength) {
  if(whiteLength >= strip.numPixels()) whiteLength = strip.numPixels() - 1;

  int      head          = whiteLength - 1;
  int      tail          = 0;
  int      loops         = 3;
  int      loopNum       = 0;
  uint32_t lastTime      = millis();
  uint32_t firstPixelHue = 0;

  set_brightness(brightness);

  while(!Serial.available()){
    for(int i=0; i<strip.numPixels(); i++) {
      if(((i >= tail) && (i <= head)) ||
        ((tail > head) && ((i >= tail) || (i <= head)))) {
        strip.setPixelColor(i, strip.Color(0, 0, 0, 255));
        strip2.setPixelColor(i, strip.Color(0, 0, 0, 255));
      }
      else{
        int pixelHue = firstPixelHue + (i * 65536L / strip.numPixels());
        strip.setPixelColor(i, strip.gamma32(strip.ColorHSV(pixelHue)));
        strip2.setPixelColor(i, strip.gamma32(strip.ColorHSV(pixelHue)));
      }
      if (Serial.available()) {     // bail out on sensor detect
            break;
      }
    }

    show_all();

    firstPixelHue += 40;
    if((millis() - lastTime) > (double)(1.0/speed)) {
      if(++head >= strip.numPixels()) {
        head = 0;
      }
      if(++tail >= strip.numPixels()) {
        tail = 0;
      }
      lastTime = millis();
    }
  }
  clear_all();
}
```

从 Neopixel Topic 读取资料，并跑 lightLED

一开始把 LED 全部设为全暗

资料读取与反馈给 ROS 知道，这边不能加 function. 不然会影响资料读取

FUNCTION 1 彩虹

设置亮度

如果有新 message 在中断 for loop 后，就中断整个 function

For loop, 走 0 – 144 次

第几颗， 什么颜色

这边只是设置彩虹色，还没亮。 目前只有两条灯条， 以后灯条多了可以加 strip3,4…

如果有新 message 中断 for loop

全亮

中断整个 function 后，清空所有 LED

```
void fillAll(int r,int g, int b, int w ,int brightness) {
  set_brightness(brightness);
  color = strip.Color(r, g, b, w);
  for(uint16_t i=0; i<strip.numPixels(); i++) {
    strip.setPixelColor(i, color);
    strip2.setPixelColor(i, color);
    if (Serial.available()) {
      break;
    }
  }
  show_all();
  while(!Serial.available()){}
  clear_all();
}
```

FUNCTION 2 全亮

设置亮度

把 RGBW 存为 color

For loop 走144次，将两条灯条的144颗 LED 设置颜色

因为灯只要设亮一次，他就会长亮，所以我们做一个空的 while loop,等新的资料进来后中断 function, 将全部 LED 清空

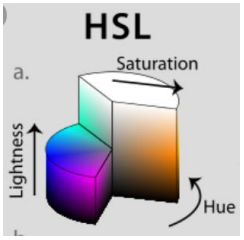
```
// Fill the dots one after the other with a color
void colorWipe(int r,int g, int b, int w ,int brightness) {
  set_brightness(brightness);
  color = strip.Color(r, g, b, w);
  while(!Serial.available()){
    if(count > strip.numPixels() || count > strip2.numPixels()){
      count=0;
      clear_all();
    }
    strip.setPixelColor(count, color);
    strip2.setPixelColor(count, color);
    count += 1;
    show_all();
  }
  clear_all();
}
```

FUNCTION 3 走马灯

```
void breath(int r,int g,int b,int w,int brightness,int startPix,int len,int speed) {
  convt.rgbToHsl(r,g,b,rgbhsl);
  set_brightness(brightness);
  int initialbright = rgbhsl[2];

  while(!Serial.available()){

    for(int i=0; i<initialbright;i+=speed){
      color = hsl(rgbhsl[0], rgbhsl[1], i);
      convt.hslToRgb(rgbhsl[0],rgbhsl[1],i,hslrgb);
      if(w!=0){
        w+=speed;
      }
      if(w>255){
        w=255;
      }
      color = strip.Color(hslrgb[0], hslrgb[1], hslrgb[2], w);
      strip.fill(color, startPix, len);
      strip2.fill(color, startPix, len);
      show_all();
      if (Serial.available()) {
        break;
      }
    }

    for(int i=initialbright; i>0;i-=speed){
      color = hsl(rgbhsl[0], rgbhsl[1], i);
      convt.hslToRgb(rgbhsl[0],rgbhsl[1],i,hslrgb);
      if(w!=0 && w>speed){
        w-=speed;
      }
      if(w<=1 && w!=0){
        w=1;
      }
      color = strip.Color(hslrgb[0], hslrgb[1], hslrgb[2], w);
      strip.fill(color, startPix, len);
      strip2.fill(color, startPix, len);
      show_all();
      if (Serial.available()) {
        break;
      }
    }
  }
  clear_all();
}
```

FUNCTION 4 两条同时呼吸灯，可以设置长度， 例如用在 停止时的呼吸灯 和后退

RGB 转 HSL (rgbhsl) 为了得到亮度 方便做 加减，rgbhsl[2] 是亮度



将亮度从 0 加到原本设置好的颜色亮度

每加一次就将 HSL 转换回 RGB (hslrgb)， 再将 加了的 W(白色亮度), 一起设置到灯条上

FUNCTION 5 左转时前后呼吸灯亮

```
void pulseL(int r,int g,int b,int w,int brightness,int startPix,int len,int speed) {

  convt.rgbToHsl(r,g,b,rgbhsl);
  strip.setBrightness(brightness);
  int initialbright = rgbhsl[2];

  for(int i=0; i<initialbright;i+=speed){
    color = hsl(rgbhsl[0], rgbhsl[1], i);
    convt.hslToRgb(rgbhsl[0],rgbhsl[1],i,hslrgb);
    if(w!=0){
      w+=speed;
    }
    if(w>255){
      w=255;
    }
    color = strip.Color(hslrgb[0], hslrgb[1], hslrgb[2], w);
    strip.fill(color, startPix, len);
    strip.fill(color, 144-len, 144);
    strip.show();
    if (Serial.available()) {
      break;
    }
  }

  for(int i=initialbright; i>0;i-=speed){
    color = hsl(rgbhsl[0], rgbhsl[1], i);
    convt.hslToRgb(rgbhsl[0],rgbhsl[1],i,hslrgb);
    if(w!=0 && w>speed){
      w-=speed;
    }
    if(w<=1 && w!=0){
      w=1;
    }
    color = strip.Color(hslrgb[0], hslrgb[1], hslrgb[2], w);
    strip.fill(color, startPix, len);
    strip.fill(color, 144-len, 144);
    strip.show();
    if (Serial.available()) {
      break;
    }
  }
}
```

这边过后需要再调 因为，前后不是对称的，如果要加多几段，就用同样的格式，只要设置 startPix 和 len 就行

这边过后需要再调 因为，前后不是对称的，如果要加多几段，就用同样的格式，只要设置 startPix 和 len 就行

```
void set_brightness(int brightness){
    strip.setBrightness(brightness);
    strip2.setBrightness(brightness);
}

void show_all(){
    strip.show();
    strip2.show();
}

void clear_all(){
    strip.clear();
    strip.show();
    strip2.clear();
    strip2.show();
}
```

```
int main(int argc,char **argv) {

  ros::init(argc, argv, "Neopixel");

  ros::NodeHandle prv_nh;
  //odom_broadcaster.reset(new tf::TransformBroadcaster);
  sub2 = prv_nh.subscribe("/joy", 50, &controlLED2);
  sub = prv_nh.subscribe("/yocs_cmd_vel_mux/output/cmd_vel", 50, &controlLED);//p2os provides the car's pose throughodometry.
  pub =prv_nh.advertise<p3dx_navigation::AdaNeopixel>("Neopixel", 100);
  ros::Rate r(10.0);
  ROS_INFO("LEDmsg2 %i",x);
  while(ros::ok())
    {
       ros::spinOnce();
       r.sleep();
    }
}
```

命名 node 为 Neopixel

Subscribe 摇杆/joy 和车子的速度 /yocs_cmd_vel_mux

发布给 Neopixel Topic

注: 其实 Node 和 Topic 名字应该不一样 避免搞混， 可是写了没办法

```
void controlLED(const geometry_msgs::Twist & msg){

    current_time = ros::Time::now();

    if(msg.angular.z<-0.25){
      // LorR,func,r,g,b,w,brightness,startPix,Len,frequency
      //array[0]= 2 (right)
      //array[0]= 3 (left)
      // switch(func):
      //case 1 :fillAll(r,g,b,w,brightness);
      //case 2: colorWipe(r,g,b,w,brightness);
      //case 3: breathAll(r,g,b,w,brightness,1);
      //case 4: whiteOverRainbow(brightness,speed,len);
      //case 5: rainbowFade2White(brightness);

      //turn left
      LEDmsg.ledcommand = {3,0,255,255,0,10,100,0,72,3};
      flag=1;
    }
    else if(msg.angular.z>0.25){
      //turn right
      LEDmsg.ledcommand = {2,0,255,255,0,10,100,0,72,3};
      flag=2;
    }
    else if(msg.linear.x<0 && msg.angular.z<0.25 && msg.angular.z>-0.25){
       //reverse
      LEDmsg.ledcommand = {0,3,150,0,0,0,100,0,34,1};
      flag=3;
    }
    else if(msg.linear.x>0 && msg.angular.z<0.25 && msg.angular.z>-0.25){
      //go straight
      LEDmsg.ledcommand = {0,1,0,50,255,0,100,0,0,1};
      flag=4;
    }
    else if(msg.linear.x==0 && msg.angular.z==0){
      //warning
      LEDmsg.ledcommand = {0,3,255,128,0,0,100,0,0,3};
      flag=5;
    }

    x=LEDmsg.ledcommand[0];
     ROS_INFO("LEDmsg %i",x);

    if(previousflag!=flag){
      if(count2==0){
      pub.publish(LEDmsg);
      }else{
        pub.publish(LEDmsg);
        sleep_until(system_clock::now() + milliseconds(100));
        pub.publish(LEDmsg);
      }
```

如果 收到车子速度资料， 将其存为 msg

如果转弯速度 小过 0.25,向左， 就设置 LEDmsg 为左转灯， 这个**如果希望 转弯更大 才发生 需要加大 号码**

为了避免重复发布 同样指令， 会设置一个 flag

当指令 不一样, flag 的号码和之前的不一样时， 才发布 message 去 arduino

第一个 message 只需要发布一次

过后就永远跑这个了，先发布一次中断清空灯条，100ms 后再发布一次让灯条亮。

```
      count2+=1;
      previousflag=flag;
    }

    last_time = ros::Time::now();

}
```

将 flag 存在 previousflag 为了比对指令有没有重复。

Lost Sync Error:
-Set in *opt/ros/melodic/lib/python-2.7/dist-packages/rosserial_python/SerialClient.py*
-set timeout

```
class SerialClient(object):
    """
        ServiceServer responds to requests from the serial device.
    """

    def __init__(self, port=None, baud=57600, timeout=100.0, fix_pyserial_for_test=False):
        """ Initialize node, connect to bus, attempt to negotiate topics. """
```

Library Installation:
https://learn.adafruit.com/adafruit-neopixel-uberguide/arduino-library-installation

Install Arduino:
sudo apt-get install ros-melodic-rosserial
sudo apt-get install ros-melodic-rosserial-arduino
download arduino IDE linux -64
cd Arduino/
rosrun rosserial_arduino make_libraries.py libraries/
upload File>>examples>>ros_lib>>ADC code to Arduino

```
rosrun rosserial_python serial_node.py /dev/ttyACM1
```