|  |  |
| --- | --- |
|  | 버튼 입력 받기 구현 |
| import RPi.GPIO as GPIO  import time  #스위치 핀 값 저장  SW = [5, 6, 13, 19]  #각 스위치별 상태 및 횟수 저장  last = [0, 0, 0, 0]  cur = [0, 0, 0, 0]  cnt = [0, 0, 0, 0]  GPIO.setwarnings(False)  GPIO.setmode(GPIO.BCM)  #핀 값을 받아와 setup 시킴  for i in range(0, 4):      GPIO.setup(SW[i], GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)  try:      while True:          for i in range(0, 4):              #스위치의 현재 상태 저장              cur[i] = GPIO.input(SW[i])              #스위치가 눌린 경우 판별              if(cur[i] == 0 and last[i] == 1):                  #스위치 눌린 횟수 저장                  cnt[i] += 1                  #스위치 및 횟수 출력                  print("('SW{} click', {})".format(i+1, cnt[i]))              #현재 상태를 이전 상태에 저장              last[i] = cur[i]  except KeyboardInterrupt:      pass  #핀 reset  GPIO.cleanup() |
|  |
| 2 | 부저로 경적 울리기 |
| import RPi.GPIO as GPIO  import time  SW = [5, 6, 13, 19]  cur = [0, 0, 0, 0]  F = [261, 294, 330, 391]  BUZZER = 12  def mel(f, n):      if(n == 1):          p.start(50)          p.ChangeFrequency(f)      elif(n==0):          p.stop()  GPIO.setwarnings(False)  GPIO.setmode(GPIO.BCM)  GPIO.setup(BUZZER, GPIO.OUT)  for i in range(0, 4):      GPIO.setup(SW[i], GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)  p = GPIO.PWM(BUZZER, 261)  try:      while True:          for i in range(0, 4):              cur[i] = GPIO.input(SW[i])              if(cur[i] == 1):                  mel(F[i], 1)                  while(GPIO.input(SW[i]) == 1): continue              mel(0, 0)  except KeyboardInterrupt:      pass  GPIO.cleanup() |
| 3 | 모터 구동하여 자동차 움직이기 |
| import RPi.GPIO as GPIO  import time  #속도 입력핀  PWM\_ = [18, 23]  #방향 입력핀  IN\_ = [22, 27, 25, 24]  #스위치 입력핀  SW = [5, 6, 13, 19]  #스위치 현재 상태  cur = [0, 0, 0, 0]  #입력받은 스위치에 따른 핀 방향 설정  num = {0: [0, 1, 0, 1], 1:[0, 1, 0, 0], 2: [0, 0, 0, 1], 3: [1, 0, 1, 0]}    GPIO.setwarnings(False)  GPIO.setmode(GPIO.BCM)  for i in range(0, 4):      #속도 제어 핀 값 setup      if i < 2 :          GPIO.setup(PWM\_[i], GPIO.OUT)          globals()["L\_Motor{}".format(i)] = GPIO.PWM(PWM\_[i],500)          globals()["L\_Motor{}".format(i)].start(0)      #방향 제어 핀 값 setup      GPIO.setup(IN\_[i], GPIO.OUT)      #스위치 핀 값 setup      GPIO.setup(SW[i], GPIO.IN, pull\_up\_down=GPIO.PUD\_DOWN)  #방향 및 속도 제어 함수  def G\_S(n, N): #(방향, 속도)      for i in range(0, 4):          #num 딕셔너리를 활용해 각 핀의 방향 설정          GPIO.output(IN\_[i], n[i])          if i < 2 :              if N == 0:                  #50%로 동작                  globals()["L\_Motor{}".format(i)].ChangeDutyCycle(50)              elif N == 1:                  #정지                  globals()["L\_Motor{}".format(i)].ChangeDutyCycle(0)  try:      while True:          for i in range(0, 4):              #스위치의 현재 상태 저장              cur[i] = GPIO.input(SW[i])              #스위치가 눌린 경우 판별              if(cur[i] == 1):                  #방향 및 속도 제어 함수 호출                  G\_S(num[i], 0)                  #스위치 출력                  print("SW{} click".format(i+1))                  #스위치 눌린 동안 무한 반복                  while(GPIO.input(SW[i]) == 1): continue              #방향 및 속도 제어 함수 호출              G\_S(num[i], 1)  except KeyboardInterrupt:      pass  #핀 reset  GPIO.cleanup() |