

2021年第10届广东省创意机器人大赛培训

编程型机器人 人工神经单元编程

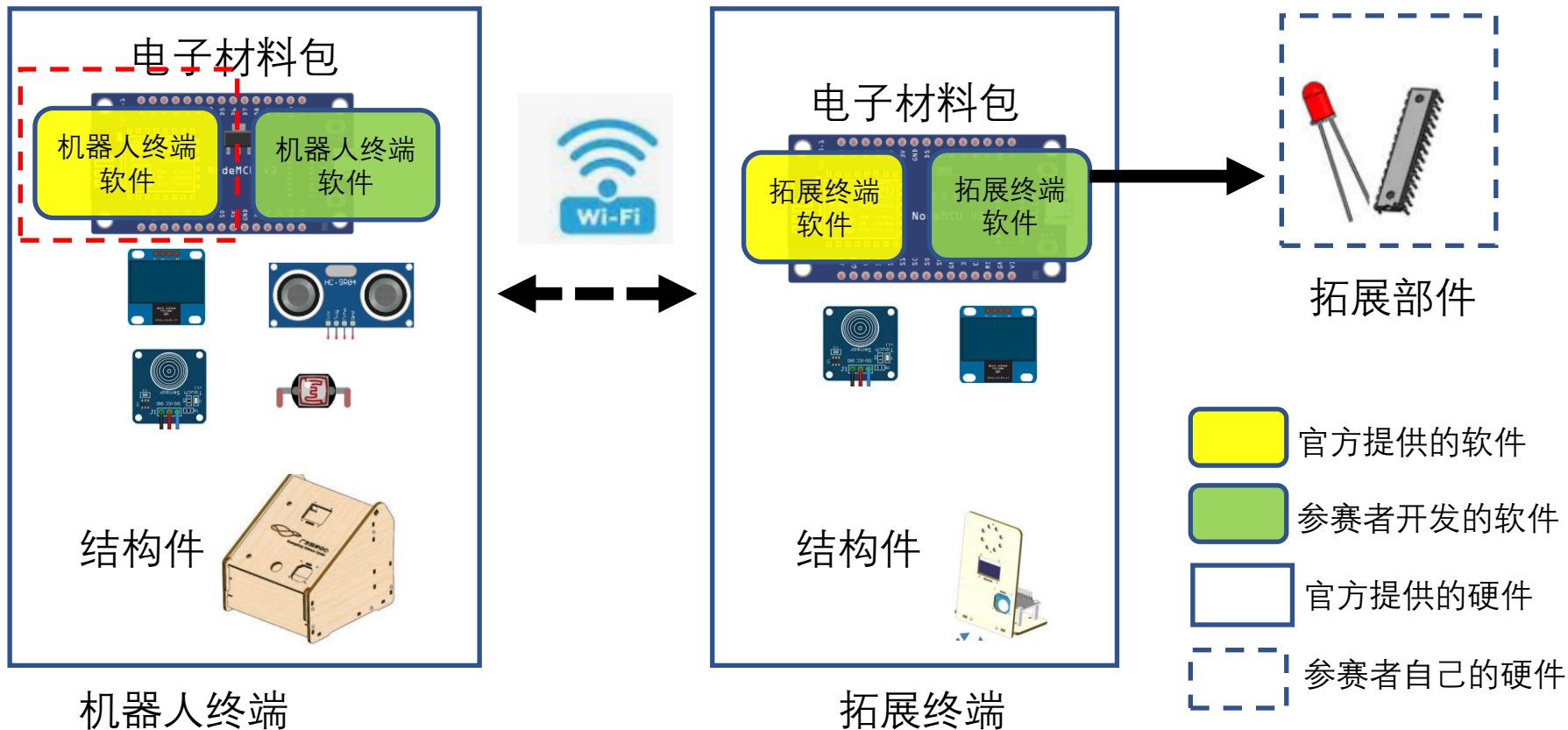
朱金辉

华南理工大学 软件学院
智能软件与机器人研究室

2021年7月

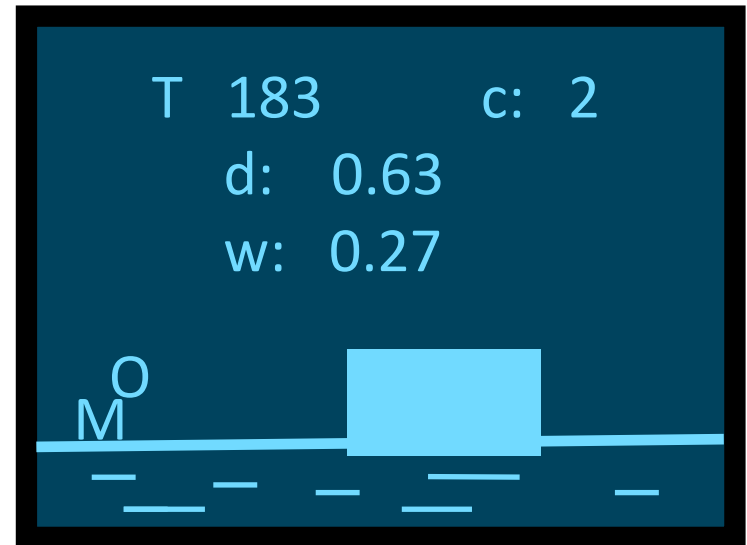
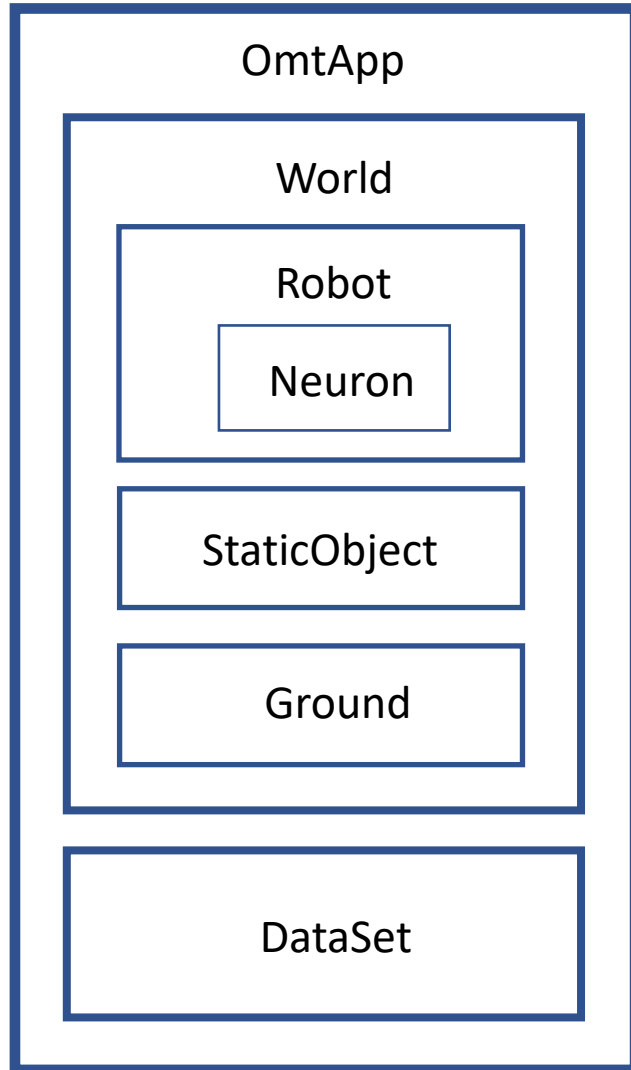


机器人的系统组成





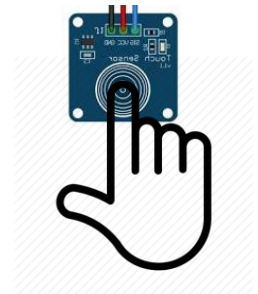
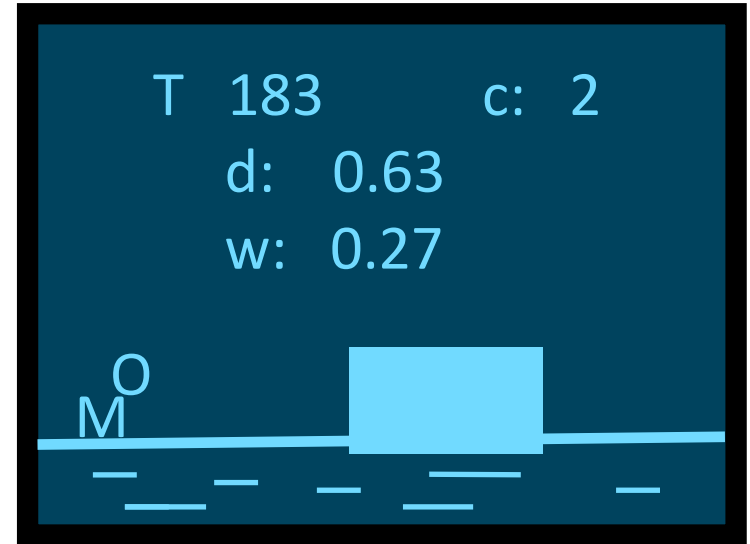
代码实现的类





机器人学习模式

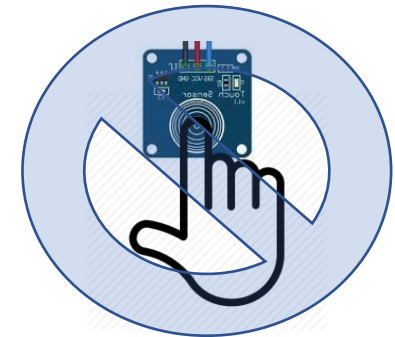
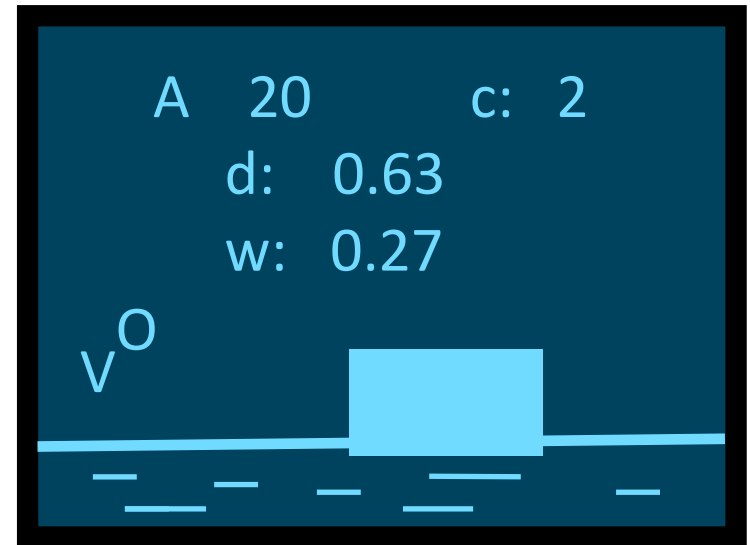
```
1 # Robot Learning
2 def run1(self):
3     self.show_mode()
4     for i in range(self.epoch1):
5         # world update
6         self.world.update()
7         # add sample
8         if(self.addSample(button.value())==True):
9             self.world.robot.n_collisions = 0
10            break # end of add samples
11        # refresh screen
12        self.show(i)
13        # sleep
14        time.sleep_ms(10)
15        # train neuron
16        train(self.dataSet,self.world.robot.neuron,0.8)
17        self.dataSet.reset()
```





AI 控制模式

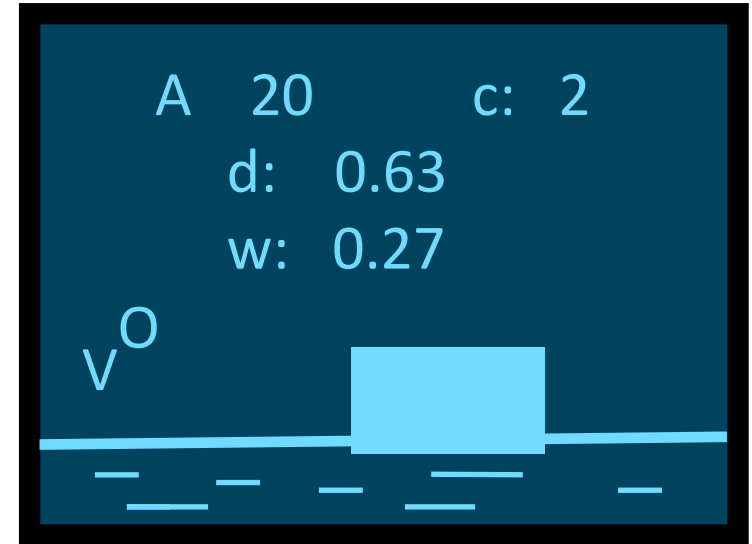
```
1 # AI control
2 def run2(self):
3     self.show_mode()
4     for i in range(self.epoch2):
5         # world update
6         self.world.update()
7         # AI decision
8         self.world.robot.think()
9         # refresh screen
10        self.show(i)
11        # sleep
12        time.sleep_ms(10)
13    ...
```





更新世界 World. update

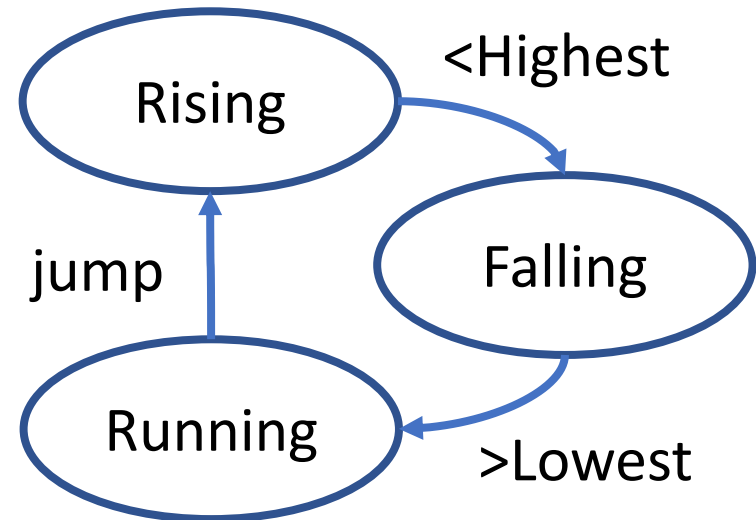
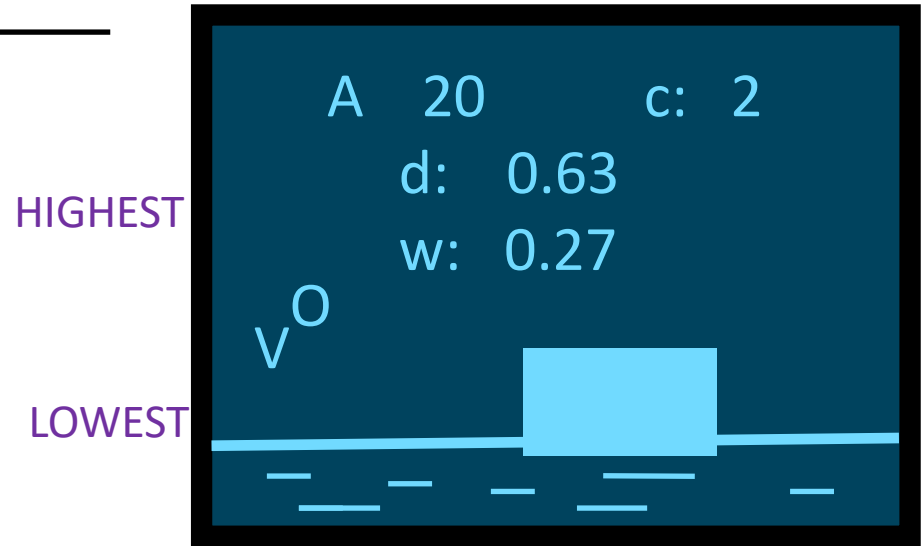
```
1 def update(self):
2     # update robot
3     self.robot.update()
4     # update x of boundary
5     self.show_x = self.robot.x - 11
6     # update ground line
7     self.ground.update(self.show_x)
8     # update obs
9     for o in range(self.n_obs):
10        self.obs[o].update(self.show_x)
11    # detection collision
12    self.collison()
13    # update distance sensor
14    self.measure()
```





更新机器人 Robot. update

```
1 def update(self):
2     self.x += self.h_speed
3
4     self.land = False
5     if(self.state == self.RISING):
6         self.y -= self.v_speed
7         if(self.y <= self.HIGHEST):
8             self.state = self.FALLING
9     elif(self.state == self.FALLING):
10        self.y += self.v_speed
11        if(self.y >= self.LOWEST):
12            self.state = self.RUNNING
13            self.land = True
14
```





机器人思考 Robot. think

```
1 def think(self):  
2     if(self.neuron.infer(self.distance,self.obs_width)):  
3         self.jump()
```

Neuron.infer

```
1 def active(self,x1,x2):  
2     self.x1 = x1  
3     self.x2 = x2  
4     s = self.w1 * self.x1 + self.w2 * self.x2 + self.b  
5     self.y = 1/(1+math.exp(-s))  
6     return self.y  
7  
8 def infer(self,x1,x2):  
9     self.active(x1,x2)  
10    return self.y>0.5
```

谢谢！

