

```

1 from pycm import *
2 import time
3 import math
4
5 class CVar:
6     #####
7     IsTorqOn                = False
8     nScreenWidth            = 0
9     nScreenHeight          = 0
10
11     nTouch_XY_X0           = 0
12     nTouch_XY_Y0           = 0
13     nTouch_Pos0            = 0
14     nTouch_Pos_X0         = 0
15     nTouch_Pos_Y0         = 0
16     nTouch_XY_X1          = 0
17     nTouch_XY_Y1          = 0
18     nTouch_Pos1           = 0
19     nTouch_Pos_X1         = 0
20     nTouch_Pos_Y1         = 0
21     nButton_0              = -1
22     nButton_1              = -1
23     nBack_Background       = 0
24
25     btn0                    = None
26     btn1                    = None
27
28     nPage                   = -1
29     nPage_Prev              = -1
30     #####
31     #fYaw_first             = 0.0
32     #fYaw                   = 0.0
33     #fYaw_turn              = 0.0
34     #####
35     nDemoMode               = 0
36     nTransform               = 0 # 0 - Humanoid, 1 - Car
37
38     # For Walking
39     nWalking                = 0
40     nWalking_Prev           = 0
41 class CTimer:
42     nTimer                   = 0
43     IsTimer                  = False
44     def __init__(self):
45         self.nTimer = 0
46         self.IsTimer = 0
47     def Set(self):

```

```

48         self.IsTimer = True
49         self.nTimer = millis()
50     def Get(self):
51         if self.IsTimer :
52             return millis() - self.nTimer
53         return 0
54     def Destroy(self):
55         self.IsTimer = False
56
57 _COLOR_NONE           = 0
58 _COLOR_WHITE         = 1
59 _COLOR_BLACK         = 2
60 _COLOR_RED           = 3

```

---

**Page 2**

```

61 _COLOR_GREEN         = 4
62 _COLOR_BLUE         = 5
63 _COLOR_YELLOW = 6
64 _COLOR_GRAY_LIGHT = 7
65 _COLOR_GRAY         = 8
66 _COLOR_GRAY_DARK = 9
67
68 _SHOW_IMAGE         = 0
69 _SHOW_TEXT         = 1
70 _SHOW_SHAPE         = 2
71 _SHOW_NUM          = 3
72
73 _RATIO              = 1000
74
75 _BTN_INDEX         = 4
76

```

77 #1..5

78 #[left,top,right,bottom, 고유 번호(ButtonNumber)] : left, top < 0 then 1..5 position

79 # Put the coordinates of each button and the unique number you want to put here-step 1/2 [Note: ButtonNumber

Do not overlap with buttons.]

```

80
81 #go button
82 _BTN_TURN_L         = [46,220,152,334,1]
83 _BTN_TURN_R         = [270,220,376,334,2]
84 _BTN_MOVE_UL        = [116,472,180,586,3]
85 _BTN_MOVE_U         = [184,420,242,532,4]
86 _BTN_MOVE_UR        = [248,472,308,586,5]
87 _BTN_MOVE_DL        = [116,720,180,828,6]
88 _BTN_MOVE_D         = [184,764,242,878,7]
89 _BTN_MOVE_DR        = [248,720,308,828,8]
90 _BTN_MOVE_L         = [90,602,156,7036,9]
91 _BTN_MOVE_R         = [274,602,336,706,10]
92
93 _BTN_MOVE_DG_FL = [70,372,130,488,11]
94 _BTN_MOVE_DG_FR = [294,372,358,488,12]
95 _BTN_MOVE_DG_BL = [70,824,130,936,13]
96 _BTN_MOVE_DG_BR = [294,824,358,936,14]
97
98
99 _BTN_CAR_MOVE_UL = [60,440,132,566,23]
100 _BTN_CAR_MOVE_U = [156,336,222,474,24]
101 _BTN_CAR_MOVE_UR = [248,440,316,566,25]

```

```

102 _BTN_CAR_MOVE_DL = [60,632,132,770,26]
103 _BTN_CAR_MOVE_D = [156,738,222,872,27]
104 _BTN_CAR_MOVE_DR = [248,632,316,770,28]
105
106 #Torque ON/OFF button
107 _BTN_TORQ = [20,38,160,160,15]
108 #Transformation button (robot->car)
109 _BTN_TRANSFORM_H2C = [766,36,970,156,17]
110 #Transformation button (car -> robot)
111 _BTN_TRANSFORM_C2H = [766,36,970,156,18]
112 #special
113 _BTN_GUN = [796,446,950,576,19]
114 _BTN_WHEEL = [796,635,950,760,20]
115
116 _PAGE_HUMANOID = 1
117 _PAGE_CAR = 2
118
119 btnList = None

```

---

### Page 3

```

120 tmrTorqBtn = CTimer()
121
122 IsPhone = False
123 Need_ReadyMotion = False
124 def ShowPage(nPage):
125     global btnList
126     CVar.nPage_Prev = CVar.nPage
127     CVar.nPage = nPage
128
129     # Create a page here and assign the button to it. -step 2/2
130     if nPage == _PAGE_HUMANOID:
131         Clear_All()
132
133         btnList = [
134             #Go button
135             _BTN_TURN_L,
136             _BTN_TURN_R,
137             _BTN_MOVE_UL,
138             _BTN_MOVE_U,
139             _BTN_MOVE_UR,
140             _BTN_MOVE_DL,
141             _BTN_MOVE_D,
142             _BTN_MOVE_DR,
143             _BTN_MOVE_L,
144             _BTN_MOVE_R,
145
146             _BTN_MOVE_DG_FL,
147             _BTN_MOVE_DG_FR,
148             _BTN_MOVE_DG_BL,
149             _BTN_MOVE_DG_BR,
150
151             #Torque ON/OFF button
152             _BTN_TORQ,
153             #Transform button (robot->car)
154             _BTN_TRANSFORM_H2C,
155             #Special action

```

```

156         _BTN_GUN,
157         _BTN_WHEEL
158     ]
159     Show_Background(1)
160 elif nPage == _PAGE_CAR:
161     Clear_All()
162
163     btnList = [
164         #Go button
165         _BTN_CAR_MOVE_UL,
166         _BTN_CAR_MOVE_U,
167         _BTN_CAR_MOVE_UR,
168         _BTN_CAR_MOVE_DL,
169         _BTN_CAR_MOVE_D,
170         _BTN_CAR_MOVE_DR,
171
172         #Torque ON/OFF button
173         _BTN_TORQ,
174         #Transform button (robot->car)
175         _BTN_TRANSFORM_C2H,
176         #Special action
177         _BTN_GUN
178     ]
179     Show_Background(2)

```

---

#### Page 4

```

180     else :
181         btnList = [
182             _BTN_TORQ
183         ]
184 def Clear_Point(nX, nY):
185     Clear(_SHOW_SHAPE, nX, nY)
186 def TorqAll(IsOn, IsSound = None):
187     TorqOnOff (-1, IsOn, IsSound)
188 def TorqOnOff(nNum, IsOn, IsSound = None):
189     if (IsOn == True) :
190         if IsSound == True :
191             buzzer.melody(14)
192         if (nNum >= 0) :
193             DXL(nNum).torque_on()
194     else :
195         dxlbus.torque_on()
196         CVar.IsTorqOn = True
197     else :
198         CVar.IsTorqOn = False
199         if IsSound == True :
200             buzzer.melody(15)
201         #Motion ends immediately
202         #Motion_Play(-3)
203         #Waiting for motion to end
204         #waitMotionStop()
205         if (nNum >= 0) :
206             540 (nnum) .torque_off ()
207     else :
208         dxlbus.torque_off()
209

```

```

210 def GetResolution():
211     for i in range(0, 100):
212         screen = smart.read32(10460)
213         CVar.nScreenWidth = screen & 0x0000FFFF
214         CVar.nScreenHeight = (screen & 0xFFFF0000) >> 16
215         if CVar.nScreenWidth > 0 and CVar.nScreenWidth < 65535 and
CVar.nScreenHeight > 0 and CVar.nScreenHeight < 65535:
216             break
217
218 def GetTouch_Down():
219     # 1 2 3 4 5
220     # 6 7 8 9 10
221     # 11 12 13 14 15
222     # 16 17 18 19 20
223     # 21 22 23 24 25
224     if (smart.is_connected() == True):
225         XY_X0 = 0
226         XY_Y0 = 0
227         Pos_X0 = 0
228         Pos_Y0 = 0
229         Pos0 = 0
230         XY_X1 = 0
231         XY_Y1 = 0
232         Pos_X1 = 0
233         Pos_Y1 = 0
234         Pos1 = 0
235
236         # Touch - First
237         Tmp = smart.read64(10470) # Touch input coordinate
238         nTouch0 = Tmp[0] & 0xffffffff

```

---

## Page 5

```

239         nTouch1 = Tmp[1] & 0xffffffff
240         IsChanged = False
241         if (nTouch0 > 0) :
242             XY_X0 = nTouch0 & 0x0000FFFF
243             XY_Y0 = (nTouch0 >> 16) & 0x0000FFFF
244             Pos_X0 = (int)((XY_X0 / CVar.nScreenWidth) * 5 + 1)
245             Pos_Y0 = (int)((XY_Y0 / CVar.nScreenHeight) * 5 + 1)
246             Pos0 = Pos_X0 + (Pos_Y0 - 1) * 5
247             XY_X0 = (int)(XY_X0 * _RATIO / CVar.nScreenWidth)
248             XY_Y0 = (int)(XY_Y0 * _RATIO / CVar.nScreenHeight)
249             if (nTouch1 > 0) :
250                 XY_X1 = nTouch1 & 0x0000FFFF
251                 XY_Y1 = (nTouch1 >> 16) & 0x0000FFFF
252                 Pos_X1 = (int)((XY_X1 / CVar.nScreenWidth) * 5 + 1)
253                 Pos_Y1 = (int)((XY_Y1 / CVar.nScreenHeight) * 5 + 1)
254                 Pos1 = Pos_X1 + (Pos_Y1 - 1) * 5
255                 XY_X1 = (int)(XY_X1 * _RATIO / CVar.nScreenWidth)
256                 XY_Y1 = (int)(XY_Y1 * _RATIO / CVar.nScreenHeight)
257
258             CVar.nTouch_Pos0 = Pos0
259             CVar.nTouch_Pos_X0 = Pos_X0
260             CVar.nTouch_Pos_Y0 = Pos_Y0
261             CVar.nTouch_XY_X0 = XY_X0
262             CVar.nTouch_XY_Y0 = XY_Y0

```

```

263
264         CVar.nTouch_Pos1 = Pos1
265         CVar.nTouch_Pos_X1 = Pos_X1
266         CVar.nTouch_Pos_Y1 = Pos_Y1
267         CVar.nTouch_XY_X1 = XY_X1
268         CVar.nTouch_XY_Y1 = XY_Y1
269     else :
270         CVar.nTouch_Pos0 = 0
271         CVar.nTouch_Pos_X0 = 0
272         CVar.nTouch_Pos_Y0 = 0
273         CVar.nTouch_XY_X0 = 0
274         CVar.nTouch_XY_Y0 = 0
275
276         CVar.nTouch_Pos1 = 0
277         CVar.nTouch_Pos_X1 = 0
278         CVar.nTouch_Pos_Y1 = 0
279         CVar.nTouch_XY_X1 = 0
280         CVar.nTouch_XY_Y1 = 0
281     else :
282         IsPhone = False
283
284 # If you enter a coordinate value between 1 and 1000, it is converted into a coordinate value suitable for a smartphone.
285 def Set(nX, nY):
286     nResX = nX * CVar.nScreenWidth / _RATIO
287     nResY = nY * CVar.nScreenHeight / _RATIO
288     return nResX, nResY
289
290 def IsButton(nX, nY, Btn):
291     if (Btn[0] < 0):
292         right = (Btn[0] * -200)
293         left = right - 200
294         bottom = (Btn [1] * -200)
295         top = bottom - 200
296         if ((nY >= top) and (nY <= bottom)):
297             if ((nX >= left) and (nX <= right)):
298                 return True
299     else:

```

---

## Page 6

```

300         if ((nY >= Btn[1]) and (nY <= Btn[3])):
301             if ((nX >= Btn[0]) and (nX <= Btn[2])):
302                 return True
303     return False
304
305 # Update motor position
306 def PositionUpdate():
307     #for i in range(0,3):
308     etc.write8(65,3)
309     while(True) :
310         if etc.read8(65) == 0 :
311             #delay(1000)
312             break
313
314 #Print number simple test
315 def Show_Num(nX, nY, nValue, nColor = _COLOR_NONE, nSize = None):
316     if (nSize == 0):
317         Clear_Num(nX, nY)

```

```

318     elif (nColor == _COLOR_NONE):
319         Show(_SHOW_NUM, nX, nY, 60, _COLOR_RED, nValue)
320     elif (nSize == None):
321         Show(_SHOW_NUM, nX, nY, 60, nColor, nValue)
322     else:
323         Show(_SHOW_NUM, nX, nY, nSize, nColor, nValue)
324 def Show_Point(nX, nY, nColor, nSize = None):
325     if (nSize == None):
326         Show(_SHOW_SHAPE, nX, nY, 20, nColor, 1)
327     else:
328         Show(_SHOW_SHAPE, nX, nY, nSize, nColor, 1)
329 def Show_Text(nX, nY, nValue, nColor = _COLOR_NONE, nSize = None):
330     if (nColor == _COLOR_NONE):
331         Show(_SHOW_TEXT, nX, nY, 60, _COLOR_RED, nValue)
332     elif (nSize == None):
333         Show(_SHOW_TEXT, nX, nY, 60, nColor, nValue)
334     else:
335         Show(_SHOW_TEXT, nX, nY, nSize, nColor, nValue)
336 def Show_Image(nX, nY, nValue, nSize = None):
337     if (nSize == None):
338         Show(_SHOW_IMAGE, nX, nY, 1, 0, nValue)
339     else:
340         Show(_SHOW_SHAPE, nX, nY, nSize, 0, nValue)
341 def Show_Background(nValue):
342     if (nValue != CVar.nBack_Background):
343         smart.display.back_image(nValue)
344         CVar.nBack_Background = nValue
345 #nShowType: 0-picture, 1-letter, 2-shape, 3-number
346 # nValue ([Image-Index], [Shape-1: Circle, 2: Square, 3: Triangle], [Text-Index], [Num-
Value])
347 # nColor (0: Unknown, 1: White, 2: Black, 3: Red, 4: Green, 5: Blue 6: Yellow, 7: Light Gray, 8: Gray,
9: dark gray)
348 def Show(nShowType, nX, nY, nSize, nColor, nValue):
349     if ((nShowType >= 0) and (nShowType < 4)):
350         nX, nY = Set (nX, nY)
351         smart.write32(10480, int(nX) | (int(nY) << 16))
352         nTmp = nValue * 256 + nSize * 65536 + nColor * 16777216
353         if (nShowType == _SHOW_IMAGE): # Do not use the color value of 0xff000000 digits
354             smart.display.front_image(nTmp & 16777215) # & 0xfffff(=16777215)
355         elif (nShowType == _SHOW_SHAPE) :
356             smart.display.shape(nTmp)
357         elif (nShowType == _SHOW_TEXT) :

```

---

## Page 7

```

358             smart.display.text(nTmp)
359         elif (nShowType == _SHOW_NUM) :
360             smart.display.number(nTmp)
361
362 def Clear(nShowType, nX, nY):
363     if ((nShowType >= 0) and (nShowType < 4)):
364         nX, nY = Set (nX, nY)
365         smart.write32(10480, int(nX) | (int(nY) << 16))
366         if (nShowType == _SHOW_IMAGE) :
367             smart.display.front_image(0)
368         elif (nShowType == _SHOW_SHAPE) :
369             smart.display.shape(0)

```

```

370         elif (nShowType == _SHOW_TEXT) :
371             smart.display.text(0)
372         elif (nShowType == _SHOW_NUM) :
373             smart.display.number(0)
374
375 def Clear_All():
376     smart.write32(10480, 0)
377     smart.display.front_image(0)
378     smart.display.shape(0)
379     smart.display.text(0)
380     smart.display.number(0)
381 def Clear_Image():
382     smart.write32(10480, 0)
383     smart.display.front_image(0)
384 def Clear_Shape():
385     smart.write32(10480, 0)
386     smart.display.shape(0)
387 def Clear_Text():
388     smart.write32(10480, 0)
389     smart.display.text(0)
390 def Clear_Num(nX = None, nY = None):
391     if (nX == None) or (nY == None) :
392         smart.write32(10480, 0)
393         smart.display.number(0)
394     else :
395         Show(_SHOW_NUM, nX, nY, 0, 0, 0)
396
397 def Move_Cover (fAngle, nSpd = 100):
398     DXL(17).write32(112, nSpd)
399     DXL (17) .write32 (116, (int) (CalcAngle2Raw (fAngle)))
400 def Move_Arm (fAngle, nSpd = 30):
401     '''
402     if fAngle >= 15:
403         fAngle = 15
404     elif fAngle <= -15:
405         fAngle = -15
406     '''
407     ValueR = (int) (CalcAngle2Raw (15 + fAngle))
408     ValueL = (int) (CalcAngle2Raw (-15 + fAngle))
409     # 448
410     DXL(13).write32(112, nSpd)
411     DXL(14).write32(112, nSpd)
412     DXL(13).write32(116, ValueR)
413     DXL(14).write32(116, ValueL)
414     return fAngle
415 # Function that converts angle to data used in motor (float -> int)
416 def CalcAngle2Raw (fAngle):
417     if fAngle == None:
418         fAngle = 0.0

```

```

419         return (int) (round (fAngle * 4096.0 / 360.0 + 2048.0))
420 # Function that converts data used in motor into angle value (int -> float)
421 def CalcRaw2Angle (nRaw):
422     if nRaw == None :
423         nRaw = 0
424     return (float) (360.0 * ((nRaw - 2048.0) / 4096.0))

```



```

425 def Get_Angle(nID):
426     return CalcRaw2Angle(DXL(nID).present_position())
427 def Test1(nBackgroundImage = 1, nImage = 0, x=500,y=500):
428     # Print background image
429     if (CVar.nTouch_Pos0 > 0):
430         # At the moment of touch, all basic shapes are erased.
431         Clear_All()
432
433     # Basic figures and drawings to hear ...
434     Show_Background(nBackgroundImage)
435     #Test Image : Show_Image(x, y, image index)
436     if (x>0) and (y>0) and (nImage>0):
437         Show_Image (x, y, nImage)
438
439     if (CVar.nTouch_Pos0 > 0):
440         Clear_Num()
441         Clear_Shape()
442         # X
443         nPos = 50
444         nGap = 20
445
446         nX = CVar.nTouch_XY_X0
447         nY = CVar.nTouch_XY_Y0
448         if nX < 100 :
449             nX = 100
450         elif nX > 900 :
451             nX = 900
452         if nY < 100 :
453             nY = 100
454         elif nY > 900 :
455             nY = 900
456
457         Show_Num (nX - nPos - nGap, nY-40, (int) (CVar.nTouch_XY_X0 / 100% 10))
458         Show_Num(nX - nPos, nY-40, (int)(CVar.nTouch_XY_X0 / 10 % 10))
459         Show_Num (nX - nPos + nGap, nY-40, (int) (CVar.nTouch_XY_X0% 10))
460         # Y
461         nPos = 50
462         Show_Num (nX + nPos - nGap, nY-40, (int) (CVar.nTouch_XY_Y0 / 100% 10))
463         Show_Num(nX + nPos, nY-40, (int)(CVar.nTouch_XY_Y0 / 10 % 10))
464         Show_Num (nX + nPos + nGap, nY-40, (int) (CVar.nTouch_XY_Y0% 10))
465
466         Show_Num(nX-30, nY+40, CVar.nTouch_Pos_X0, _COLOR_GREEN)
467         Show_Num(nX+30, nY+40, CVar.nTouch_Pos_Y0, _COLOR_GREEN)
468
469         Show_Point(CVar.nTouch_XY_X0, CVar.nTouch_XY_Y0, _COLOR_BLUE)
470
471     if (CVar.nTouch_Pos1 > 0):
472         # X
473         nPos = 50
474         nGap = 20
475
476         nX = CVar.nTouch_XY_X1
477         nY = CVar.nTouch_XY_Y1
478         if nX < 100 :

```

```

479         elif nX > 900:
480             nX = 100
481             nX = 900
482         if nY < 100 :
483             nY = 100
484         elif nY > 900 :
485             nY = 900
486
487         Show_Num (nX - nPos - nGap, nY-40, (int) (CVar.nTouch_XY_X1 / 100% 10))
488         Show_Num(nX - nPos, nY-40, (int)(CVar.nTouch_XY_X1 / 10 % 10))
489         Show_Num (nX - nPos + nGap, nY-40, (int) (CVar.nTouch_XY_X1% 10))
490         # Y
491         nPos = 50
492         Show_Num (nX + nPos - nGap, nY-40, (int) (CVar.nTouch_XY_Y1 / 100% 10))
493         Show_Num(nX + nPos, nY-40, (int)(CVar.nTouch_XY_Y1 / 10 % 10))
494         Show_Num (nX + nPos + nGap, nY-40, (int) (CVar.nTouch_XY_Y1% 10))
495
496         Show_Num(nX-30, nY+40, CVar.nTouch_Pos_X1, _COLOR_GREEN)
497         Show_Num(nX+30, nY+40, CVar.nTouch_Pos_Y1, _COLOR_GREEN)
498
499         Show_Point(CVar.nTouch_XY_X1, CVar.nTouch_XY_Y1, _COLOR_RED)
500
501 #Sets the motor's operating speed (Based on Position): 0 initialization
502 def Setup_Speed(nID, nValue) :
503     DXL(nID).write32(112, nValue) # 112 : profile velocity
504 def Motion_Ready(IsOpened=False):
505     # All motor torque On
506     TorqAll(True)
507     # Update the entire motor position
508     PositionUpdate()
509     #delay(100)
510     if (IsOpened) :
511         Motion_Play_And_Wait(40)
512     else :
513         Motion_Play_And_Wait(1)
514 def Motion_Stop():
515     Motion_Play(-3)
516 def Motion_Play(nMotionIndex, nNextMotion = 0):
517     if (nMotionIndex <= 0):
518         # Stop operation (0)
519         #nMotionIndex => -3: End immediately, -2: End after executing step(Key-Frame), -1:page(motion
Unit) execution and exit, 0: page (motion unit) execution and exit after execution.
520         motion.play((int)(nMotionIndex))
521     else:
522         Setup_Speed(254, 0)
523         if (nNextMotion == 0) :
524             motion.play((int)(nMotionIndex))
525         else :
526             motion.play((int)(nMotionIndex), (int)(nNextMotion))
527 def Motion_Play_And_Wait(nMotionIndex, nNextMotion = 0):
528     Motion_Play(nMotionIndex, nNextMotion)
529     Motion_Wait()
530 def Motion_Wait() :
531     while motion.status():
532         if btnList != None:
533             # Check button press-consumption of touch
534             nNum0, nNum1, Event_Dn0, Event_Dn1, Event_Up0, Event_Up1, Btn0, Btn1 =
GetButton(btnList)
535 def WaitButtonUp():
536     while(True) :

```

```
537         # Check button press
538         nNum0, nNum1, Event_Dn0, Event_Dn1, Event_Up0, Event_Up1, Btn0, Btn1 =
GetButton(btnList)
539         if ((nNum0 < 0) and (nNum1 < 0)):
540             break
541
542 def GetButton(btnList):
543     #Check the touch input of the smartphone
544     GetTouch_Down()
545
546     nNum0 = -1
547     nNum1 = -1
548     nDown0 = 0
549     nDown1 = 0
550     nUp0 = 0
551     nUp1 = 0
552     Btn0 = None
553     Btn1 = None
554     nnum = 0
555     nCnt = 0
556     if (CVar.nTouch_Pos0 > 0) :
557         nCnt = nCnt + 1
558     if (CVar.nTouch_Pos1 > 0) :
559         nCnt = nCnt + 1
560
561     nPass = 0
562     for btn in btnList:
563         if (CVar.nTouch_Pos0 > 0):
564             if (IsButton(CVar.nTouch_XY_X0, CVar.nTouch_XY_Y0, btn) == True) :
565                 nNum0 = btn [_BTN_INDEX]
566                 Btn0 = btn
567                 nPass = nPass | 0x01
568         if (CVar.nTouch_Pos1 > 0):
569             if (IsButton(CVar.nTouch_XY_X1, CVar.nTouch_XY_Y1, btn) == True) :
570                 nNum1 = btn [_BTN_INDEX]
571                 Btn1 = btn
572                 nPass = nPass | 0x10
573         if (nCnt == 1) :
574             if (nPass > 0):
575                 break
576         else:
577             if (nPass == 0x11):
578                 break
579         nnum + 1 = nnum
580
581     if ((nNum1 == nNum0) and (nNum0 >= 0)):
582         nNum1 = -1
583
584     # switching
585     #IsSwitching = False
586     if (((nNum0 >= 0) and (nNum0 == CVar.nButton_1)) or ((nNum1 >= 0) and (nNum1 ==
CVar.nButton_0))) :
587         nNum2 = nNum1
588         nNum1 = nNum0
```

```
589         nNum0 = nNum2
590         #IsSwitching = True
591     #Button Down Event
592     if (nNum0 >= 0):
593         if (CVar.nButton_0 != nNum0):
594             nDown0 = 1
595             CVar.btn0 = Btn0
```

---

## Page 11

```
596     else :
597         if (CVar.nButton_0 >= 0):
598             nUp0 = 1
599     if (nNum1 >= 0):
600         if (CVar.nButton_1 != nNum1):
601             nDown1 = 1
602             CVar.btn1 = Btn1
603     else :
604         if (CVar.nButton_1 >= 0):
605             nUp1 = 1
606
607     CVar.nButton_0 = nNum0
608     CVar.nButton_1 = nNum1
609     if nUp0 == 1:
610         Btn0 = CVar.btn0
611     if nUp1 == 1:
612         Btn1 = CVar.btn1
613
614     return nNum0, nNum1, nDown0, nDown1, nUp0, nUp1, Btn0, Btn1
615
616 def Wheel(nSpd_L, nSpd_R):
617     nGap = 0
618
619     nGap_L = nGap
620     nGap_R = nGap
621     if (nSpd_L < 0):
622         nGap_L = -10
623     if (nSpd_R < 0):
624         nGap_R = -10
625     etc.write8(1200,0)
626     etc.write16(1202,108)
627     etc.write8(1204,4)
628     #
629     nID = 15
630     etc.write8 (1205, nID)
631     etc.write32(1206,0)
632     etc.write8(1200,1)
633     #
634     nID = 16
635     etc.write8 (1205, nID)
636     etc.write32(1206,0)
637     etc.write8(1200,1)
638     #
639     etc.write8(1200,2)
640     #####
641     etc.write8(1200,0)
642     etc.write16(1202,104)
643     etc.write8(1204,4)
```

```

644 #
645 nID = 15
646 etc.write8 (1205, nID)
647 etc.write32(1206,-nSpd_R + nGap_R)
648 etc.write8(1200,1)
649 #
650 nID = 16
651 etc.write8 (1205, nID)
652 etc.write32 (1206, nSpd_L + nGap_L)
653 etc.write8(1200,1)
654 #
655 etc.write8(1200,2)
656 def Demo0():

```

---

## Page 12

```

657 nValue = 128 # Gunshot
658 nOctave = 3
659 nScale = 3
660 instrument_and_scale = nValue * 65536 + nOctave * 256 + nScale
661 #smart.multimedia.play_instrument(instrument_and_scale)
662
663 if (CVar.nTransform == 0):
664     Motion_Play_And_Wait(4,5)
665
666     Motion_Play(5,6)
667     delay(800)
668     smart.multimedia.play_instrument(instrument_and_scale)
669     Motion_Wait()
670
671     Motion_Play(6,7)
672     delay(800)
673     smart.multimedia.play_instrument(instrument_and_scale)
674     Motion_Wait()
675
676     Motion_Play(7)
677     delay(800)
678     smart.multimedia.play_instrument(instrument_and_scale)
679     Motion_Wait()
680
681     Motion_Ready()
682 else:
683     smart.multimedia.play_instrument(instrument_and_scale)
684 def Demo1():
685     nWheelValue = 30
686     Wheel(nWheelValue,nWheelValue)
687     Motion_Play_And_Wait(8,9)
688
689     delay(1000)
690
691     nWheelValue = 240
692     Wheel(nWheelValue,nWheelValue)
693
694     Motion_Play(9)
695     smart.multimedia.play_audio1(1)
696
697     delay(6000)
698

```

```

699     Wheel(70,70)
700
701     Move_Cover(90, 60)
702
703     delay(1000)
704     Wheel(20,20)
705     smart.multimedia.play_audio1(2)
706     delay(1000)
707     Wheel(0,0)
708     Motion_Ready(True)
709 def Move_Turn(fAngle, nSpd, IsWait = True):
710     Set_Rot()
711     if fAngle <0:
712         nSpd = -nSpd
713     nSpeed = nSpd
714     nSlow = nSpd
715     if (abs(nSpd) > 25):
716         if (fAngle <0):
717             nSlow = -25

```

---

## Page 13

```

718         else:
719             nSlow = 25
720     #Set the Inposition value by speed difference.
721     fInposition = 1
722     fSlowPosition = 10
723     if (IsWait):
724         while(True):
725             Wheel(-nSpeed, nSpeed)
726             diff = Get_Rot()
727             if (abs(diff) >= abs(fAngle) - fInposition):
728                 break
729             # From this section, the speed decreases.
730             if (abs(diff) >= abs(fAngle) - fSlowPosition):
731                 nSpeed = nSlow
732             else:
733                 nSpeed = nSpd
734         Move_Stop()
735     else:
736         Wheel(-nSpeed, nSpeed)
737 def Move2 (fAngle = 0.0, spd = 100, dist = 500):
738     Move_Arm (fAngle)
739     Move (Cos (fAngle), Sin (fAngle), spd, dist)
740 def Move_Stop():
741     DXL(15).write32(104, 0)
742     DXL(16).write32(104, 0)
743 def Move(f = 1.0, w = 0.0, spd = 100, dist = 500):
744     # Back and forth
745     fX=round(f,3)
746     # Right and left
747     fY=round(w,3)
748     # Height (let's just use speed here)
749     fZ=spd
750
751     IsForward = True
752     if (f < 0):
753         IsForward = False

```

```

754         fX = -fX
755         fY = -fY
756
757     dir = 0
758     if (fY < 0) :
759         dir = -1
760     elif (fY > 0):
761         dir = 1
762     V=fX*fZ
763     # Circle diameter
764     width=140
765     # Distance from outer wheel to rotation point
766     R=dist*abs(fY) + width/2
767
768     VL=V*(1-dir*width/(2*R))
769     VR=V*(1+dir*width/(2*R))
770     V=fZ
771     VL2=(fX-fY)*V
772     VR2=(fX+fY)*V
773
774     t1 = VL + VL2
775     t2=VR+VR2
776     if (IsForward == False):
777         t1 = -t1

```

---

**Page 14**

```

778         t2 = -t2
779     #print(t1, t2)
780     Wheel(round(t1), round(t2))
781 def CheckButton(Btn0, Btn1, Btn) :
782     if (Btn0 == Btn) :
783         return 1
784     elif (Btn1 == Btn):
785         return 2
786     return 0
787 def CheckTouch_for_walking(nDirection):
788     CVar.nWalking = 0
789     if ((nDirection >= 11) and (nDirection <= 13)) :
790         # 11 (forward-left) 12 (forward) 13 (forward-right)
791         CVar.nWalking = 1
792     elif ((nDirection >= 21) and (nDirection <= 23)):
793         # 21 (Backward-left) 22 (Backward) 23 (Backward-right)
794         CVar.nWalking = -1
795 def walking(nDirection) :
796     CVar.nWalking = 0
797     if (nDirection > 0):
798         CheckTouch_for_walking(nDirection)
799         nMotion = 0
800         nMotion_Next = 0
801         if (CVar.nWalking != 0):
802             nMotion = 10
803             if (CVar.nWalking == -1):
804                 nMotion += 10
805             nMotion_Next = nMotion + 2
806             Motion_Play(nMotion, nMotion_Next)
807

```

```

808 nCnt = 0
809 CVar.nWalking_Prev = CVar.nWalking
810 while(CVar.nWalking != 0) :
811     while(motion.status() == True):
812         # Control mode
813         nDirection = 0
814         if (btnList != None):
815             nNum0, nNum1, Event_Dn0, Event_Dn1, Event_Up0, Event_Up1,
Btn0, Btn1 = GetButton(btnList)
816             if CheckButton(Btn0, Btn1, _BTN_MOVE_UL) :
817                 nDirection = 11
818             if CheckButton(Btn0, Btn1, _BTN_MOVE_U) :
819                 nDirection = 12
820             if CheckButton(Btn0, Btn1, _BTN_MOVE_UR) :
821                 nDirection = 13
822             if CheckButton(Btn0, Btn1, _BTN_MOVE_DL) :
823                 nDirection = 21
824             if CheckButton(Btn0, Btn1, _BTN_MOVE_D) :
825                 nDirection = 22
826             if CheckButton(Btn0, Btn1, _BTN_MOVE_DR) :
827                 nDirection = 23
828             CheckTouch_for_walking(nDirection)
829         if (CVar.nWalking != 0):
830             if (CVar.nWalking != CVar.nWalking_Prev):
831                 CVar.nWalking = CVar.nWalking_Prev
832                 break
833             nCnt = 1
834         else :
835             nCnt = 0
836         if (nCnt > 0) :
837             nCnt = nCnt - 1

```

---

**Page 15**

```

838         else :
839             break
840
841         # [F]ready: 10, left: 11, go: 12, right: 13, end: 14
842         # [B]ready: 20, left: 21, go: 22, right: 23, end: 24
843
844         nMotion = nDirection
845         nMotion_Next = nMotion
846         Motion_Play(nMotion, nMotion_Next)
847         CVar.nWalking_Prev = CVar.nWalking
848         if (CVar.nWalking_Prev != CVar.nWalking):
849             CVar.nWalking = CVar.nWalking_Prev
850
851         nMotion = 14
852         if (CVar.nWalking < 0):
853             nMotion += 10
854         Motion_Play_And_Wait(nMotion)
855         Motion_Ready()
856
857
858
#####
#####
859 console(USB)

```



```

860 #console(BLE)
861 #console(UART)
862
863 # controller direction : 0-vertical(Humanoid), 1-Horizontal
864 eeprom.imu_type(0)
865 # Turn off the torque and modify the actuator setting and controller setting according to the robot.
866 TorqAll(False)
867 # profile -> velocity-based
868 DXL(254).write8(10, 0) # 0 -> velocity-based profile, 4 -> time-based profile
869 # Secondary ID(255: No Use, 0 ~ 252: ID)
870 DXL(254).write8(12, 255) # 255 -> No Use(Clear)
871 # Operation Mode(1:velocity[wheel], 3:position)
872 DXL(254).mode(3) # position
873 # Wheel Setting
874 DXL(15).mode(1) # velocity
875 DXL(16).mode(1) # velocity
876
877 TorqAll(True)
878
879 # In actual use, nTest = 0 should be used.
880 nTest = 0 # 0-Normal, 1-Coordinate output
881 nTest_BackgroundImage = 1
882
883 # Return all motor speeds to initial state
884 Setup_Speed(254, 0)
885
886 if (nTest == 0):
887     Motion_Ready()
888 while(True) :
889     if (IsPhone == False) :
890         if (smart.is_connected() == True):
891             #Check if it is connected to the smartphone.
892             smart.wait_connected()
893             #Set the smartphone screen horizontally. (0: Auto, 1: Vertical, 2: Horizontal)
894             smart.display.screen_orientation(2)
895             # Waiting for the screen to switch before getting the screen's width and height.

```

---

## Page 16

```

896         delay(500)
897
898         # Get the resolution of the screen and put it in CVar.nScreenWidth, CVar.nScreenHeight.
899         GetResolution()
900
901         #Print background image
902         ShowPage(_PAGE_HUMANOID)
903
904         # Writes the smartphone connection to the variable
905         IsPhone = True
906     else :
907         # Test 1 => Coordinate output
908         if (nTest == 1) :
909             #Check the touch input of the smartphone
910             GetTouch_Down()
911             Test1(nTest_BackgroundImage)
912         else: #Run
913             # Check button press

```

```

914 nNum0, nNum1, Event_Dn0, Event_Dn1, Event_Up0, Event_Up1, Btn0, Btn1 =
GetButton(btnList)
915
916 IsDown = True
917 if (Event_Dn1 == 1):
918     smart.etc.vibrate(10)
919 elif (Event_Dn0 == 1):
920     smart.etc.vibrate(10)
921 else:
922     IsDown = False
923
924 if (CVar.nTransform == 0):
925
926
927     #####
928     # Control mode
929     nDirection = 0
930     if CheckButton(Btn0, Btn1, _BTN_MOVE_UL) :
931         nDirection = 11
932     if CheckButton(Btn0, Btn1, _BTN_MOVE_U) :
933         nDirection = 12
934     if CheckButton(Btn0, Btn1, _BTN_MOVE_UR) :
935         nDirection = 13
936     if CheckButton(Btn0, Btn1, _BTN_MOVE_DL) :
937         nDirection = 21
938     if CheckButton(Btn0, Btn1, _BTN_MOVE_D) :
939         nDirection = 22
940     if CheckButton(Btn0, Btn1, _BTN_MOVE_DR) :
941         nDirection = 23
942     walking(nDirection)
943     #####
944
945
946     if (Btn0 == _BTN_TURN_L) or (Btn1 == _BTN_TURN_L):
947         Motion_Play_And_Wait(34)
948         Need_ReadyMotion = True
949     if (Btn0 == _BTN_TURN_R) or (Btn1 == _BTN_TURN_R):
950         Motion_Play_And_Wait(35)
951         Need_ReadyMotion = True
952
953     if (Btn0 == _BTN_MOVE_DG_FL) or (Btn1 == _BTN_MOVE_DG_FL):
954         Motion_Play_And_Wait(30)

```

```

955         Need_ReadyMotion = True
956     if (Btn0 == _BTN_MOVE_DG_FR) or (Btn1 == _BTN_MOVE_DG_FR):
957         Motion_Play_And_Wait(31)
958         Need_ReadyMotion = True
959     if (Btn0 == _BTN_MOVE_DG_BL) or (Btn1 == _BTN_MOVE_DG_BL):
960         Motion_Play_And_Wait(32)
961         Need_ReadyMotion = True
962     if (Btn0 == _BTN_MOVE_DG_BR) or (Btn1 == _BTN_MOVE_DG_BR):
963         Motion_Play_And_Wait(33)
964         Need_ReadyMotion = True
965
966     if (Btn0 == _BTN_MOVE_L) or (Btn1 == _BTN_MOVE_L):

```

```

967         Motion_Play_And_Wait(36)
968         Need_ReadyMotion = True
969     if (Btn0 == _BTN_MOVE_R) or (Btn1 == _BTN_MOVE_R):
970         Motion_Play_And_Wait(37)
971         Need_ReadyMotion = True
972
973     if ((nNum0 < 0) and (nNum1 < 0)):
974         if Need_ReadyMotion == True:
975             Motion_Ready()
976             Need_ReadyMotion = False
977     else:
978         Need_ReadyMotion = False
979         nSpeed = 3
980         nSpd = round(250 * nSpeed / 5 / 2)
981         IsMoving = False
982         nX = 1
983         nY = 0.5
984         fAngle = 20
985     if (Btn0 == _BTN_CAR_MOVE_UL) or (Btn1 == _BTN_CAR_MOVE_UL):
986         Move_Arm (-fAngle)
987         Move(nX,nY,nSpd, 300)
988         IsMoving = True
989     if (Btn0 == _BTN_CAR_MOVE_U) or (Btn1 == _BTN_CAR_MOVE_U):
990         Move_Arm(0)
991         Move(nX,0,nSpd, 0)
992         IsMoving = True
993     if (Btn0 == _BTN_CAR_MOVE_UR) or (Btn1 == _BTN_CAR_MOVE_UR):
994         Move_Arm (fAngle)
995         Move(nX,-nY,nSpd, 300)
996         IsMoving = True
997     if (Btn0 == _BTN_CAR_MOVE_DL) or (Btn1 == _BTN_CAR_MOVE_DL):
998         Move_Arm (-fAngle)
999         Move(-nX,-nY,nSpd, 300)
1000
1001         IsMoving = True
1002
1003     if (Btn0 == _BTN_CAR_MOVE_D) or (Btn1 == _BTN_CAR_MOVE_D):
1004
1005         Move_Arm(0)
1006
1007         Move(-nX,0,nSpd, 300)
1008
1009         IsMoving = True
1010
1011     if (Btn0 == _BTN_CAR_MOVE_DR) or (Btn1 == _BTN_CAR_MOVE_DR):
1012
1013         Move_Arm (fAngle)
1014
1015         Move(-nX,nY,nSpd, 300)
1016
1017
1018
1019         IsMoving = True
1020
1021     if (IsMoving == False):
1022
1023         Move_Stop()

```

```

101_1
101
2         if (IsDown):
101
3             if (Btn0 == _BTN_GUN) and (Event_Dn0):
101
4                 Demo0()
101
5             if (Btn0 == _BTN_WHEEL) and (Event_Dn0):
101
6                 Demo1()
101
7             #if CVar.IsWalking == True and IsWalking == False:
101
8             # CVar.IsWalking = False
101
9             # Motion_Ready()
102
0
102
1             #Special action
102
2             #_BTN_GUN,
102
3             #_BTN_WHEEL
102
4
102
5             #Transform button (robot->car)(car->robot)
102
6             if (Btn0 == _BTN_TRANSFORM_H2C) or (Btn1 == _BTN_TRANSFORM_H2C):
102
7                 CVar.nTransform = ( CVar.nTransform + 1 ) % 2
102
8                 Motion_Play_And_Wait(3 - CVar.nTransform)
102
9
103
0                 ShowPage(CVar.nTransform + 1)
103
1             elif (Btn0 == _BTN_TRANSFORM_C2H) or (Btn1 == _BTN_TRANSFORM_C2H):
103
2                 CVar.nTransform = ( CVar.nTransform + 1 ) % 2
103
3                 Motion_Play_And_Wait(3 - CVar.nTransform)
103
4
103
5                 Demo0()
103
6                 Demo1()
103
7                 #Motion_Ready()

```



106

7

106

8

106

9

107

0

107

1

107

2

107

3

107

4

buzzer.melody(1)

#TorqOff variable

CVar.IsTorqOn = False

# No more timer detection.

tmrTorqBtn.Destroy()