

IOT BASED SAFETY GADGET FOR CHILD **SAFETY MONITORING & NOTIFICATION**

TEAM ID: PNT2022TMID27560

FINAL CODE

```
import json

import wiotp.sdk.device import time myConfig = {
    "identity": {
        "ordId": "n7q4jv",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}

client = wiotp.sdk.device.DeviceClient(config=myConfig, loghandlers=None)
client.connect()

while True:
    name= "Smartbridge"
    #in area location

    latitude= 17.42225176
```

```
longitude= 78.5458842
```

```
#out area location
```

```
#latitude= 17.4219272
```

```
#logitude= 78.5488783
```

```
myData={'name': name, 'lat':latitude, 'lon':longitude}
```

```
client.publishEvent(eventId="status",msgFormat="json", data=myData, qos=0, onPublish=None)
```

```
print("Data published to IBM IoT platform: ",myData)
```

```
time.sleep(5)
```

```
client.disconnect()
```

```
import time
```

```
def stopwatch(seconds,d,lspoin t):
```

```
start = time.time()
```

```
time.clock()
```

```
elapsed = 0
```

```
flag =False
```

```
num = 0 while elapsed < seconds:
```

```
elapsed = time.time() - start
```

```
print "%02d" % elapsed if elapsed > d[num] and elapsed< d[num+1] and flag == False:x =  
lspoint[num][0] y = lspoint[num][1] createpoint(x,y)
```

```
flag = True print "Shot Taken" print point_in_poly(x,y,polygon)
```

```
if elapsed > d[num+1]:
```

```
print "Shot Taken" flag == False
```

```
num = num+1
```

```
x = lspoint[num][0]
```

```

y = lspoint[num][1] createpoint(x,y) print
point_in_poly(x,y,polygon) time.sleep(1)

def createpoint(x,y):

crs = "point?crs=epsg:27700&field=id:integer"

layer= QgsVectorLayer(crs, 'points' , "memory")

pr= layer.dataProvider()

pt = QgsFeature() point1 = QgsPoint(x,y) pt.setGeometry(QgsGeometry.fromPoint(point1))
pr.addFeatures([pt])      # update extent of the layer

layer.updateExtents() # add the second point pt = QgsFeature()

QgsMapLayerRegistry.instance().addMapLayers([layer]) def point_in_poly(x,y,poly): n =
len(poly) inside = False p1x,p1y = poly[0] for i in range(n+1):

p2x,p2y = poly[i % n]if y > min(p1y,p2y):

if y <= max(p1y,p2y):

if x <= max(p1x,p2x): if p1y != p2y:

p1x)/(p2y-p1y)+p1x

xints = (y-p1y)*(p2x-

if p1x == p2x or x <=

xints:inside = not inside p1x,p1y = p2x,p2y

return inside

#### define the polygon polygon =

[(512882.78819722467,120811.83924772343),(512960.84437170526,120809.7007223952),(51
2960.
84437170526,120809.7007223952),(512959.77510904113,120754.09906386107),(512882.788
19722 467,120756.2375891893)]

```

```

#### set how long the script will run (70 seconds will get you in and out of geofence)
time_seconds = 70 #### first coordinate x = 512915 y = 120728

#### time intervals, 10 seconds between shots / or points intervals = int(time_seconds / 10)
lspoint = []

#### build the list of coordinates to be plotted for i in range(0,intervals+1):

y1 = y + (i*12.5) lspoint.append([x,y1])


#### to build the blocks of time in intervals, so we know the number of intervals (default is 7),

#### we need a list of time intervals [0,10,20,30 etc] to check against the clock this list is d, f is
the gap ie 10 seconds, a is starting point (0)

### b is the number of intervals + 1 because the code will check the the next in the list f = 10 a
= 0 b = intervals+1 d = [x * f for x in range(a, b)]

### Run the stopwatch, or start the program! stopwatch(time_seconds,d,lspoint)

```