3ʳᵈ party webot robot controllers in python IDLE

My Computer > Properties > Advanced System Settings > Environment Variables >

System Variable	Value
PYTHONPATH	C:\Python27\Lib;C:\Python27\DLLs;C:\Python27\Lib\lib-tk;C:\Program Files\Webots\lib\controller\python27;
WEBOTS_HOME	C:\Program Files\Webots
PYTHONIOENCODING	UTF-8
Path (add these)	C:\Python27\;C:\Python27\Scripts\;C:\Program Files\Webots\lib\controller\python27;C:\Program Files\Webots\msys64\mingw64\bin;C:\Program Files\Webots\lib\controller;${WEBOTS_HOME}\lib\controller;

Webots world

Set the robot controller to **<extern>** , make a note of the robot name field

Sample robot controller (save and run in IDLE)

```
from controller import Robot, DistanceSensor, Motor       leftMotor.setPosition(float('inf'));rightMotor.setPosition(float('inf'))
TIME_STEP = 12; robot = Robot()                           leftMotor.setVelocity(4); rightMotor.setVelocity(-4)

leftMotor = robot.getMotor('motor1')                      while robot.step(TIME_STEP)!=-1:
rightMotor = robot.getMotor('motor2')                       pass
```

Add this before
WEBOTS_ROBOT_NAME = "bot one"
wb_robot_init()

Internal python Webots controllers

Using complex libraries
libraries like thread, math, sys import without problems
import Tkinter fails from a webots native python controller. It cannot find the necessary library
make a new directory webots/lib/tcl/Python27 then copy c:/python27/tcl into that directory
then tkinter will run from a native webots controller...

Supervisor tasks in webots python
Sample robot controller (save and run in WEBOTS)

```
from controller import Supervisor, Motor       def reset:
robot = Supervisor()                               robot.simulationSetMode(2) #crashes unless in fast mode
                                                   robot.simulationReset() #restart the simulation
#using supervisor instead of robot                 robot.simulationSetMode(1) #back to normal play mode
#supervisor has all the functions of Robot too
                                                while robot.step(12)!=-1: pass
```

This allows more streamlined control of the robot without having to manually re-rerun an IDE or webots every time. Robot position can be done from within the controller without reloading the controller
although thread does work...it gives a stdout error