

# DESIGN ENGINEERING: ELECTRONICS 2 – SYSTEM DESIGN

## Group 7

Anna Bernbaum, Clarisse Bret, Felix Crowther, Benedict Greenberg

### README for project work

For videos and pictures please use this link:

<http://bit.ly/DE2-EPDS-G7-MEDIA>

Final tuning values for Pybench-09

$K_P = 5.41$ ,  $K_I = 0.22$ ,  $K_D = 0.33$

Included files:

<b>g7_balance_beat.py</b>	Switch setting: 00 Program will self-balance robot and detect beat using blue LED as output. This is the main self-balancing program.
g7_balance_bluetooth.py	Switch setting: 10 Example program of how Bluetooth can be implemented whilst balancing. Beat detection also occurring here. Program works although can cause the robot to fall if there is too much interference from user.
<b>g7_balance_dance.py</b>	Switch setting: 01 Program demonstrating balancing robot reacting to beat detection with attempted dance moves.
g7_balance.py	Program used to tune K values for PID control.
g7_choreo_functions.py	Program used to translate commands from txt file to motor instructions.
g7_choreo.txt	First choreography produced from MATLAB analysis.
g7_choreo2.txt	Second choreography produced from MATLAB analysis.
<b>g7_dance.py</b>	Switch setting: 11 Program that runs routine based on beat detection. This runs without balancing robot due to complex and rapid moves.
g7_drive.py	Simple example of Bluetooth control of robot before balancing or dancing.
g7_pid_controller.py	Class which runs PID control on Segway for balancing.
main.py	Modified main script to utilise the binary switch on PyBench.

In addition, we used the standard modules provided:

- boot.py
- drive.py
- font.py
- main.py
- mic.py
- motor.py
- mpu6050.py
- oled\_938.py
- pybench\_main.py
- pybench\_test.py
- pybench.py
- user.py