

Running a Custom Python Script at Startup in PYNQ

This document explains how to add a custom Python script to a PYNQ image and configure it to run automatically at boot.

Step1: Create the Package Directory

- * Create the package folder:

```
mkdir -p PYNQ/sdbuild/packages/hello_world
```

Step2: Create the Python Script

- * Create the file:

```
vi PYNQ/sdbuild/packages/hello_world/hello-world.py
```

- * Add the following content:

```
#!/usr/bin/env python3  
print("Hello, World!")
```

- * Make it executable using the below command

```
chmod +x PYNQ/sdbuild/packages/hello_world/hello-world.py
```

Step3: Create the Systemd Service

- * Create the service file:

```
vi PYNQ/sdbuild/packages/hello_world/bootscript.service
```

- * Add the following content:

```
[Unit]  
Description=Run hello-world at boot  
  
[Service]  
Type=oneshot  
ExecStart=/bin/bash -c '/usr/local/bin/hello-world > /dev/ttyPS0'  
RemainAfterExit=yes  
  
[Install]  
WantedBy=multi-user.target
```

Step4: Add the Package Install Script

- * Create the file:

```
vi PYNQ/sdbuild/packages/hello_world/pre.sh
```

- * Add the following content:

```
#!/bin/bash
```

```
set -e
```

```
target=$1
```

```
script_dir="$( cd "$( dirname "${BASH_SOURCE[0]}" )" && pwd )"
```

```
sudo mkdir -p $target/usr/local/bin
```

```
sudo install -m 0755 $script_dir/hello-world.py $target/usr/local/bin/hello-world
```

```
sudo mkdir -p $target/etc/systemd/system
```

```
sudo install -m 0644 $script_dir/bootscript.service $target/etc/systemd/system/bootscript.service
```

```
sudo mkdir -p $target/etc/systemd/system/multi-user.target.wants
```

```
sudo ln -sf /etc/systemd/system/bootscript.service \
```

```
    $target/etc/systemd/system/multi-user.target.wants/bootscript.service
```

- * Make it executable using the below command

```
chmod +x PYNQ/sdbuild/packages/hello_world/pre.sh
```

Step5: Update the board specification file:

- * Edit the board specification file and the line provided below

```
vi PYNQ/boards/ig60m/ig60m.spec
```

- * Add the following line:

```
STAGE4_PACKAGES_ig60m += hello_world
```

Step6: Build the Image

- * From the sdbuild directory, rebuild the board image:

```
cd PYNQ/sdbuild
```

```
make BOARDS=ig60m
```

- * This rebuilds the final PYNQ image and includes the custom package in the root filesystem.

- * Refer to Section 2.2.3 of the Software User Guide for full build details

Step7: Verify on the Target board

- * After flashing/booting the generated image, log into the target and run:

```
root@pynq:/home/xilinx# hello-world
```

```
Hello, World!
```