PyCNC 是一款免费的开源高性能 G 代码解释器和 CNC/3D 打印机控制器。它可以在各种基于 Linux 的 ARM 主板上运行，例如 Raspberry Pi、Odroid、Beaglebone 等。这让您可以灵活地选择您最熟悉的主板，并使用 Linux 提供的所有功能，同时将所有 G 代码运行时保留在同一主板上，而无需使用单独的微控制器进行实时操作。与 C/C++ 项目相比，我们选择 Python 作为主要编程语言，大大减少了代码库，减少了样板和微控制器特定的代码，并使更广泛的受众可以修改该项目。

Linux 中的实时电机控制?

通常，由于缺乏实时 GPIO 控制，因此无法从 Linux 运行时环境控制步进电机。即使基于内核的模块也无法保证对步进电机脉冲的精确控制。但是，我们可以使用单独的硬件模块 DMA（直接内存访问），它为 GPIO 输出提供高精度。该模块可以使用基于主芯片内部振荡器的时钟将代表 GPIO 状态的字节从 RAM 缓冲区直接复制到 GPIO，而无需使用 CPU 内核。使用这种方法，该项目为移动步进电机生成脉冲，而且无论 CPU 负载和 OS 时间抖动如何，这种方法都非常精确。
这种方法还允许将 Python 语言用于该项目。通常，Python 不是实时应用程序的最佳选择，但由于项目只需要设置 DMA 缓冲区，硬件将完成其余工作，因此 Python 成为轻松开发该项目的完美选择。

视频演示 - YouTube 视频
以及 PyCNC 还只是原型时的原始视频YouTube 视频

当前的 gcode 和功能支持

- 支持 G0、G1、G2、G3、G4、G17、G18、G19、G20、G21、G28、G53、G90、G91、G92、M2、M3、M5、M30、M84、M104、M105、M106、M107、M109、M114、M140、M190 等命令。可轻松添加命令，请参阅 gmachine.py 文件。
- 支持四轴——X、Y、Z、E。
- 支持XY、ZX、YZ平面的圆弧插补。
- 支持带转速控制的主轴。
- 支持挤出机和床加热器。
- 硬件看门狗。

看门狗

PyCNC 使用其中一个 DMA 通道作为硬件看门狗，以确保安全。如果主板、操作系统或 PyCNC 挂起，此看门狗应在 15 秒内禁用所有 GPIO 引脚（通过将它们切换到输入状态，对于 RPi，这将是 GPIO0-29）。由于存在高电流和危险设备（如加热床、挤出机加热器），此功能应可防止无法控制的过热。但不要过分依赖此类软件功能，它们也可能挂起或输出 MOSFET 短路，请在机器中使用硬件保护，如热切断开关。

硬件

目前，该项目支持 Raspberry Pi 1-3。使用 RPI3 开发和测试。并且有一种方法可以添加新板。请参阅hal.py文件。
注意：当前 Raspberry Pi 实现使用与板载 3.5 毫米插孔（PWM 模块）相同的资源，因此不要使用它。HDMI 音频有效。

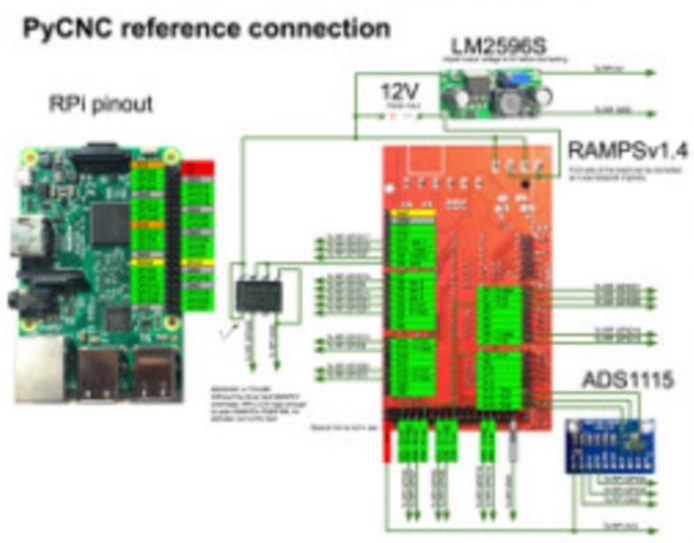
配置

所有配置都存储在config.py中，包含硬件属性、限制和硬件控制的引脚名称。Raspberry Pi 实现应连接到 A4988、DRV8825 或任何其他具有 DIR 和 STEP 引脚输入的步进电机驱动器。为 Raspberry Pi 2-3 创建了默认配置，并且此接线配置：

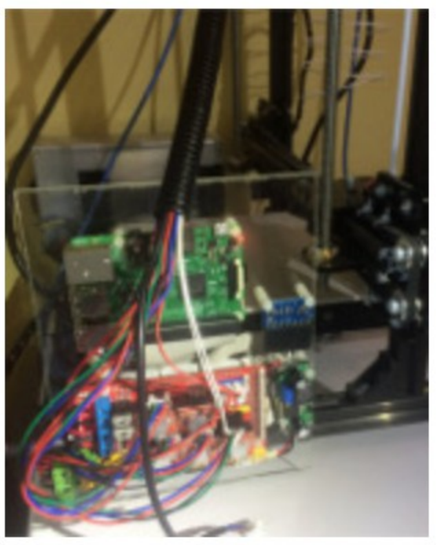
电路名称	RPi 引脚名称	RAMPSv1.4 板引脚名称	笔记
X步	GPIO21	A0	
X 目录	GPIO20	A1	
步进器使	GPIO26	A2、A8、D24、D30、D38	所有步进机
Y 步	GPIO16	A6	
Y 方向	GPIO19	A7	
方向	GPIO13	D48	
Z 步骤	GPIO12	D46	
E1 步骤	GPIO6	D36	预订
E1 目录	GPIO5	D34	预订
E0 目录	GPIO7	D28	
E0步骤	GPIO8	D26	
最大值	GPIO11	D19	
Z 最小值	GPIO25	D18	
最大	GPIO9	D15	
Y 最小值	GPIO10	D14	
最大X	GPIO24	D2	
X 分钟	GPIO23	D3	
加热床	GPIO22	D8	
加热器2	GPIO27	D9	风扇的用途
加热器 1	GPIO18	D10	
1 系列	GPIO17	D11	预订
2 号	GPIO15	D6	预订
3 星	GPIO4	D5	预订
4 星	GPIO14	D4	预订
I2C 时钟源	GPIO3	-	到 ads111x
I2C 数据总线	GPIO2	-	到 ads111x
ads1115 ch0	-	A15	加热器 2 - nc
ads1115 ch1	-	A14	床传感器

ads1115 ch2	-	A13	挤出机传感器
ads1115 ch3	-	-	未连接

因此，通过这种方式连接 Raspberry Pi，无需为项目配置引脚图。RAMPS v1.4板可用于此目的。完整的参考电路图和组装控制器的照片（点击放大）：







用法

只需克隆此 repo 并`./pycnc`从 repo 根目录运行。它将以交互式终端模式启动，其中可以手动输入 gcode 命令。
要使用 gcode 命令运行文件，只需运行`./pycnc filename`。
也`pycnc`可以选择安装。运行

```
sudo pip install .
```

在 repo 根目录中安装它。之后，`pycnc`命令将添加到系统路径。要删除安装，只需运行：

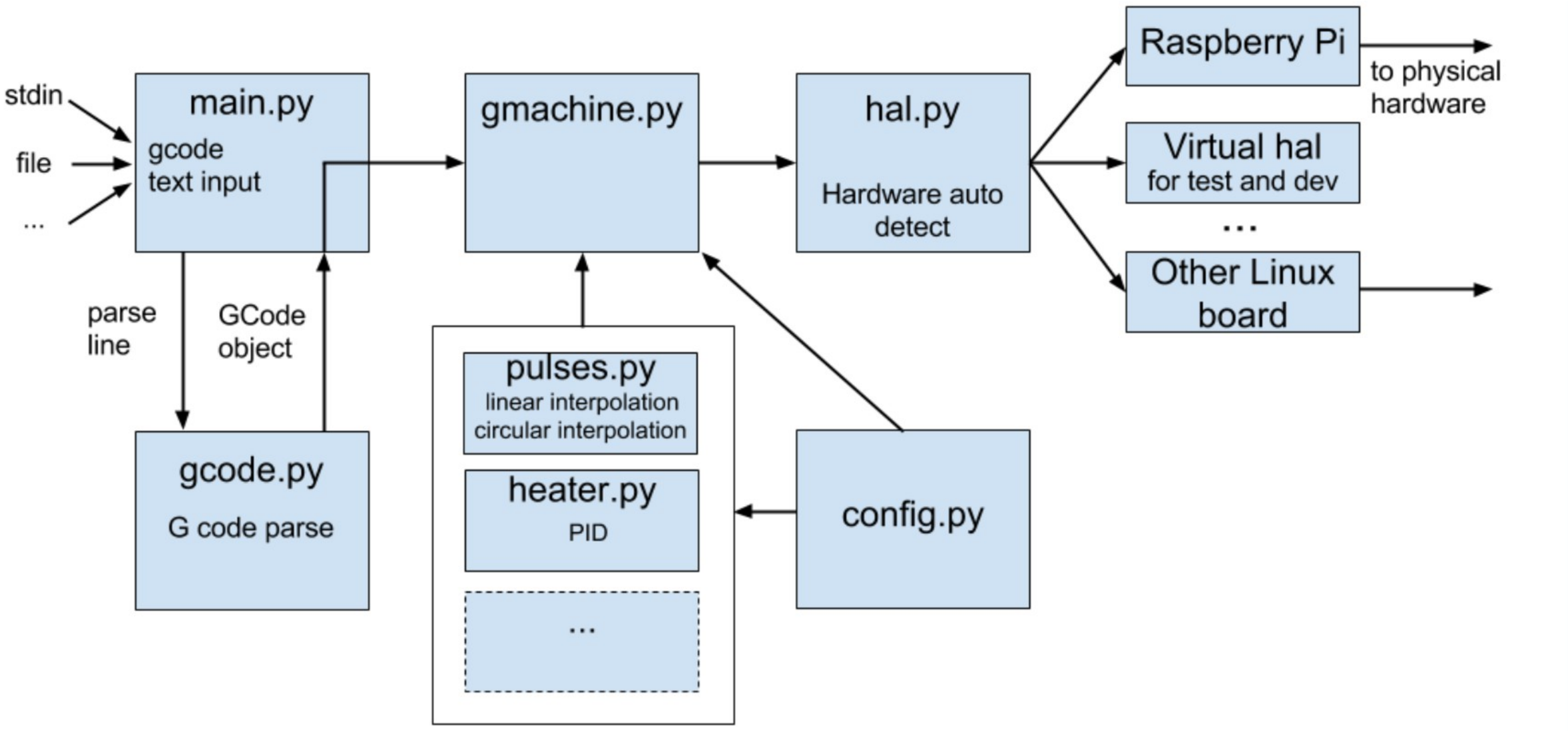
```
sudo pip remove pycnc
```

绩效通知

纯 Python 解释器无法为高速机器提供出色的性能。超速设置会导致电机误脉冲，并可能丢失轨迹。根据我的测试，Raspberry Pi 2 可以处理 400 个脉冲的轴，最高速度约为每分钟 800 毫米。总会有出路的！:) 使用像 PyPy 这样的 JIT Python 实现。使用 PyPy，RPi2 可以在机器上处理每毫米 80 步的电机，速度高达每分钟 18000 毫米。
注意：Raspbian 在存储库中的 PyPy 版本已过时（v4.0）。此外，v4.0 在模块实现方面存在问题。使用 PyPy v5.0+，从此处`mmap`为您的操作系统下载。PyPy
安装：

```
wget wget https://bitbucket.org/pypy/pypy/downloads/pypy2-v5.7.1-linux-armhf-raspbian.tar.bz2
sudo mkdir /opt/pypy
sudo tar xvf pypy2-v5.7.1-linux-armhf-raspbian.tar.bz2 --directory /opt/pypy/ --strip-componen
sudo ln -s /opt/pypy/bin/pypy /usr/local/bin/pypy
```

项目架构



依赖项

运行时无需任何操作。只有纯 Python 代码。要上传到 PyPi，需要`pandoc`：

```
sudo dnf install pandoc
sudo pip install pypandoc
```

GCode 模拟

只是一个链接，主要供我自己使用 :)，一个用于 gcode 文件模拟的不错的 Web 软件（对于手动创建 gcode 文件非常有帮助）： https://nraynaud.github.io/webgcode/

执照

参见LICENSE文件。

作者

尼古拉·哈巴罗夫