Toolchain installation

Tobias Tefke[1], Prof. Ralf C. Staudemeyer[1]

[1] Schmalkalden University of Applied Sciences, Schmalkalden, Germany

November 14, 2025

To be able to program the PineCone Internet of Things (IoT) devices, you have to install a corresponding Software Development Kit (SDK) first. In this document, it is shown how this toolchain is installed on GNU/Linux.

1. First of all, compilers and interpreters for multiple programming languages must be installed:

```
sudo apt-get install build-essential python3 python3-pip git screen wget xz-
    ↪ utils curl libssl-dev libbz2-dev ncurses-dev libffi-dev libsqlite3-dev
    ↪ tk-dev liblzma-dev libreadline-dev device-tree-compiler
```

2. Furthermore, we have to set up the *pyenv* tool to manage a virtual python environment:

```
curl -fsSL https://pyenv.run | bash

echo 'export PYENV_ROOT="$HOME/.pyenv"' >> ~/.bashrc
echo '[[ -d $PYENV_ROOT/bin ]] && export PATH="$PYENV_ROOT/bin:$PATH"' >> ~/.
    ↪ bashrc
echo 'eval "$(pyenv init - bash)"' >> ~/.bashrc
```

3. Next, the code of the SDK has to be cloned:

```
git clone --recursive git@github.com:ttefke/bl602_iot_sdk.git
```

   Cloning using HTTP (discouraged):

```
git clone --recursive https://github.com/ttefke/bl602_iot_sdk.git
```

4. Afterwards, set the environment variables for the SDK:

```
cd bl602_iot_sdk
echo export BL60X_SDK_PATH=$(pwd) >> ~/.bashrc
echo export CONFIG_CHIP_NAME=BL602 >> ~/.bashrc
```

5. Then, create a directory for the compiler and get it:

```
mkdir -p toolchain/compiler
wget https://github.com/ttefke/riscv-gnu-toolchain/releases/download
    ↪ /2025.08.29/suas-gcc-15-1.tar.xz -O /tmp/toolchain.tar.xz
tar xJf /tmp/toolchain.tar.xz -C toolchain/compiler
```

6. Download the flashing tool and make it executable:

```
mkdir -p toolchain/bin
wget https://github.com/ttefke/blflash/releases/download/suas-0.1/blflash -O toolchain/bin/blflash
chmod u+x toolchain/bin/blflash
```

7. Add the bin directory to your PATH variable:

```
echo export PATH="\$PATH:$(pwd)/toolchain/bin" >> ~/.bashrc
```

8. Allow the user to access serial ports:

```
sudo usermod -a -G dialout $USER
```

9. Now restart your shell so that all changes come into effect:

```
exec "$SHELL"
```

10. Create a virtual python environment:

```
pyenv install 3.12
pyenv virtualenv 3.12 bl_venv
```

11. Now, activate the python environment (you have to repeat this step for each shell you open):

```
pyenv activate bl_venv
```

12. Finally, you can compile and flash the hello world app to test your application. Please see the ‘Compiling and flashing’ document to find the according instructions.