

# How to Cross Compile PRU code from the Host Machine

By Team PlantPal: Chris Aloise, Gurman Chauhan, Hieu Duong, Osman Saleem

Last Update: Apr 18, 2024

This guide will walk you through the process of setting up your host PC to cross-compile PRU (Programmable Realtime Unit) code for the BeagleBone. This will allow you to compile your PRU code on your host machine, instead of having to copy it onto the target board (BBG) and compile it there, making your development process more efficient and streamlined.

## Formatting

1. Commands for the host Linux console are shown as:  
`(host)$ echo "Hello PC world!"`
2. Commands for the target (BeagleBone) Linux console are shown as:  
`(bbg)$ echo "Hello embedded world!"`
3. All commands are case-sensitive.

## Prerequisites

- A host PC running a Linux distribution. This guide was tested on Debian 11.8.
- Internet connection to download necessary tools
- Basic knowledge of command-line operations in Linux

## Table of Contents

Step 1: Install the PRU Code Generation Tools (CGT).....	2
Step 2: Install the PRU Software Support Package (PSSP).....	3
Step 3: Compile Your PRU Code.....	3
Conclusion.....	3
Resources.....	4
Automated Install Script.....	4





## Step 1: Install the PRU Code Generation Tools (CGT)

The [PRU CGT](#) is a set of utilities that includes a compiler, assembler, linker, and other tools necessary for developing PRU applications.

**WARNING:** Do NOT move on to the next step until the current one is finished successfully.

1. Download the latest version of the tools from the [Texas Instruments website](#). We will be using the **Linux Installer for PRU CGT** version **2.3.3** for this guide.

### Downloads

↓ <a href="#">Windows Installer for PRU CGT</a> — 15837 K	Windows Installer for PRU CGT
	MD5 checksum <code>c80d6fb40f00e870417bdb81a18c1ced</code> 
↓ <a href="#">MacOS Installer for PRU CGT</a> — 17919 K	MacOS Installer for PRU CGT
	MD5 checksum <code>8c129a5d09f009b3ac34a4c79e9ac91c</code> 
↓ <a href="#">Linux Installer for PRU CGT</a> — 17442 K	Linux Installer for PRU CGT
	MD5 checksum <code>abb76fac986993aafaf467915985ec4f</code> 
↓ <a href="#">ARM-A8 Installer for PRU CGT</a> — 40689 K	ARM-A8 Installer for PRU CGT
	MD5 checksum <code>648a6d7d8162fd6a89f381c7b974e6b0</code> 

2. Navigate to the downloaded directory and make the file executable:

```
(host)$ chmod +x ti_cgt_pru_2.3.3_linux_installer_x86.bin
```

3. Run the installer as root:

```
(host)$ sudo ./ti_cgt_pru_2.3.3_linux_installer_x86.bin
```

### Troubleshooting:

- If you encounter: `sudo: unable to execute ...: No such file or directory`, install `lib32z1` and try again.

```
(host)$ sudo apt install lib32z1
```

4. Follow the prompts to complete the installation.

When prompted for **Destination Directory**, enter: `/usr/share/ti/cgt-pru`

5. Link to the system path:

```
(host)$  
for file in /usr/share/ti/cgt-pru/bin/*; do  
    filename="$(basename "$file")"  
    sudo ln -sf "$file" "/usr/bin/$filename"  
done
```

## Step 2: Install the PRU Software Support Package (PSSP)

The [PSSP](#) includes libraries and example code for developing PRU applications. We will be using version **6.4.0** for this guide.

1. Clone the PSSP repository:

```
(host)$ sudo git clone --depth 1 --branch v6.4.0  
git://git.ti.com/pru-software-support-package/pru-software-  
support-package.git  
/usr/lib/ti/pru-software-support-package-v6.4.0
```

2. Link to the system path:

```
(host)$ sudo ln -sf  
/usr/lib/ti/pru-software-support-package-v6.4.0  
/usr/lib/ti/pru-software-support-package
```

## Step 3: Compile Your PRU Code

1. Navigate to the directory containing your PRU code.

**Note:** Make sure your directory does not contain any spaces in its name, the compiler is very finicky about this.

2. Run the PRU compiler as you normally would with a Makefile:

```
(host)$ make
```

3. If the compilation is successful, you should see an output file with the `.out` extension in your current directory under the `gen` folder.

## Conclusion

You should now be able to cross-compile PRU code from your host machine. This can make your development process more efficient, as you don't have to copy your code onto the BeagleBone and compile it there. Happy coding!

## Resources

- [PRU Guide by Dr. Brian Fraser](#)
- [PRU Cookbook by Mark A. Yoder](#)
- [TI PRU-CGT Technical Documentation](#)

## Automated Install Script

An [installation script](#) is provided for your convenience to save time and streamline the process. The script is a live document and a static version is included below.

```
#!/bin/bash

# Exit script on first error
set -e

num_warnings=0

# -----
# Do NOT run this on the BeagleBone.
if [ -f "/ID.txt" ]; then
    echo "ERROR: Do NOT run this on the BeagleBone."
    exit 1
fi

# Do NOT install on a path with spaces in its name.
if [[ $0 == *" "* ]]; then
    echo "ERROR: The path to this script contains spaces."
    echo "Please move this script to a path without spaces and try again."
    exit 1
fi

# Ensure we are on x86_64
if [ "$(uname -m)" != "x86_64" ]; then
    echo "ERROR: Host needs to be x86 to use the CGT installer."
    exit 1
fi

#-----
```

```

# Create a temporary directory
# See: https://stackoverflow.com/a/34676160

TEMP_DIR=$(mktemp -d)

# check if tmp dir was created
if [[ ! "$TEMP_DIR" || ! -d "$TEMP_DIR" ]]; then
    echo "Could not create temp dir"
    exit 1
fi

# deletes the temp directory
function cleanup {
    rm -rf "$TEMP_DIR"
    echo "Deleted temp working directory $TEMP_DIR"
}

# register the cleanup function to be called on the EXIT signal
trap cleanup EXIT

#-----
# TI Code Generation Tools (CGT) for PRU
readonly PRU_CGT_VER="2.3.3"
readonly
PRU_CGT_BIN="ti_cgt_pru_${PRU_CGT_VER}_linux_installer_x86.bin"
readonly PRU_CGT_BIN_PATH="$TEMP_DIR/$PRU_CGT_BIN"
readonly
PRU_CGT_URL="https://software-dl.ti.com/codegen/esd/cgt_public_sw/PRU
/$PRU_CGT_VER/$PRU_CGT_BIN"
readonly CGT_PRU_PATH="/usr/share/ti/cgt-pru"

# Download
sudo apt-get install -y curl lib32z1
echo "Downloading $PRU_CGT_URL"
curl -L -o "$PRU_CGT_BIN_PATH" "$PRU_CGT_URL"

# Install
chmod +x "$PRU_CGT_BIN_PATH"
sudo "$PRU_CGT_BIN_PATH" --mode unattended --prefix /usr/share/ti

```

```

# Create symbolic links
# Use -f to overwrite
if [ ! -d "$CGT_PRU_PATH" ]; then
    sudo ln -sf "/usr/share/ti/ti-cgt-pru_${PRU_CGT_VER}"
"$CGT_PRU_PATH"
else
    echo "WARNING: $CGT_PRU_PATH already exists."
    num_warnings=$((num_warnings + 1))
fi

for file in "$CGT_PRU_PATH"/bin/*; do
    filename=$(basename "$file")
    sudo ln -sf "$file" "/usr/bin/$filename"
done

# -----
# PRU Software Support Package (PSSP)
readonly PSSP="pru-software-support-package"
readonly PSSP_VER="v6.4.0"
readonly PSSP_URL="git://git.ti.com/$PSSP/$PSSP.git"
readonly PSSP_PATH="/usr/lib/ti/$PSSP"
readonly PSSP_VER_PATH="$PSSP_PATH-$PSSP_VER"

# Download
if [ ! -d "$PSSP_VER_PATH" ]; then
    echo "Cloning $PSSP_URL"
    sudo git clone --depth 1 --branch "$PSSP_VER" "$PSSP_URL"
"$PSSP_VER_PATH"
else
    echo "WARNING: $PSSP_VER_PATH already exists."
    num_warnings=$((num_warnings + 1))
fi

# Create symbolic link
# Use -f to overwrite
if [ ! -d "$PSSP_PATH" ]; then
    sudo ln -sf "$PSSP_VER_PATH" "$PSSP_PATH"
else

```

```
    echo "WARNING: $PSSP_PATH already exists."
    num_warnings=$((num_warnings + 1))
fi

# -----
echo "==> SUCCESS: Installed PRU tools with $num_warnings warnings."
```