

AM335x Processor SDK RTOS 2.0.0.0

How to modify the UART0 example to enable UART1 on a BeagleBone Black

Introduction

This document describes the steps needed to modify the default UART0 example in the AM335x Processor SDK RTOS package, to enable UART1. On the BeagleBone Black P9 header, pins 24(TX) and 26(RX) are connected to UART1. This procedure will also show a test to verify that UART1 is enabled.

Tutorial environment

- CCSV6.1
- Windows 7 Host
- Processor SDK RTOS AM335x 2.0.0.0 install path C:\ti\pdk_am335x_1_0_0\packages
- [BeagleBone Black](#) Rev C (BBB) with a debugger header solder to P2. Here is a [video](#) on attaching a header to P2.
- [Blackhawk USB100v2](#)

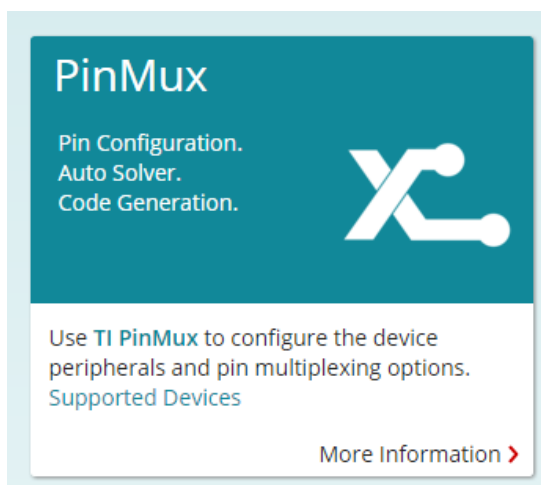
Pre-requisites

- [Download/Install the AM335x Processor SDK RTOS package for Windows host.](#)
- After installing the package, create the **MyExampleProjects** by running the pdkProjectCreate.bat file described [here](#) under “PDK Example and Test Project Creation”. Please pay attention to your CCS install location definition in the .bat file. Ensure that the new packages installed are registered with CCS and do not have CCS running in the background when executing the .bat file.

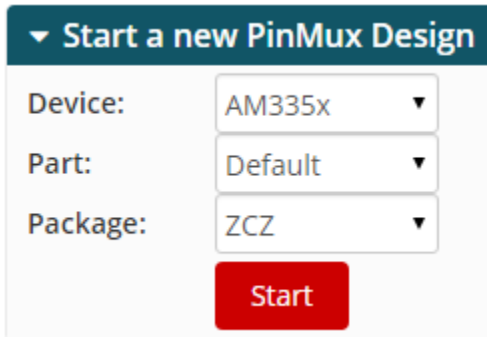
Software Modifications

PinMux utility

This procedure will use the [cloud-based pinmux utility](#)

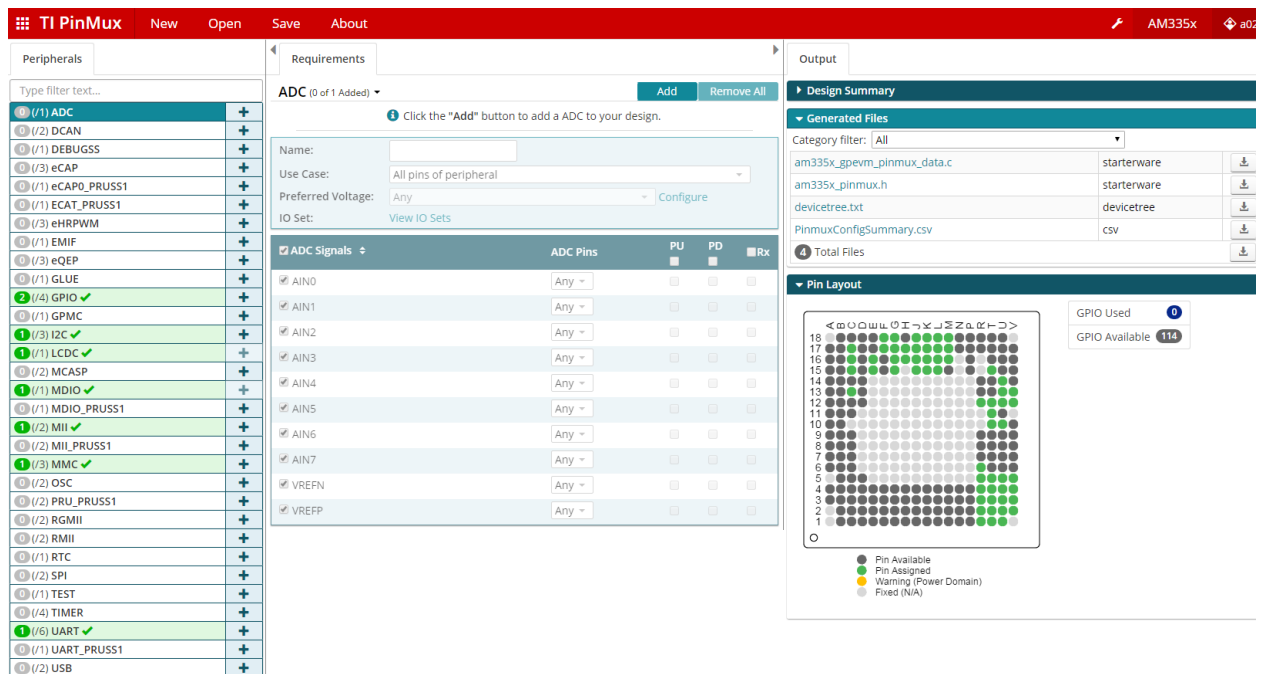


- 1) Select the below parameters and click start.



- 2) Once the pinmux utility opens, a default pinmux configuration for the BBB would need to be loaded into the tool. This file is beaglebone_black_config found under
C:\ti\pdk_am335x_1_0_0\packages\ti\starterware\tools\pinmux_config\am335x

Open>Local Drive and navigate to the above path, load beaglebone_black_config.
At this point you will see the default pinmux configuration loaded.



- 3) Add UART 1 to the configuration. The default config only has UART0. Click the “+” symbol at the UART row



- 4) This will create the new UART instance in the requirements section

UART (2 of 6 Added) Add Remove All

- ✓ UART 0
- ✗ MyUART3

Name: MyUART3

Use Case: All pins of peripheral

Use Peripheral: Any(UART1)

Preferred Voltage: Any Configure

IO Set: UART1_IOSet_2 View IO Sets

| UART Signals | UART Pins | PU | PD | Rx |
|--------------------|----------------------------------|--------------------------|--------------------------|-------------------------------------|
| ✓ ctsn(uart1_ctsn) | Any(D18) | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| ✓ dcdn | Any Pin conflict | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| ✓ dsrn | Any Pin conflict | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| ✓ dtrn | Any Pin conflict | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| ✓ rin | Any Pin conflict | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| ✓ rtsn(uart1_rtsn) | Any(D17) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| ✓ rxd(uart1_rxd) | Any(D16) | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| ✓ txd(uart1_txd) | Any(D15) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

5) Change the description to match the fields below

UART (2 of 6 Added) Add Remove All

- ✓ UART 0
- ✓ UART 1

Name: UART 1

Use Case: UART with RXD and TXD only

Use Peripheral: UART1 🔒

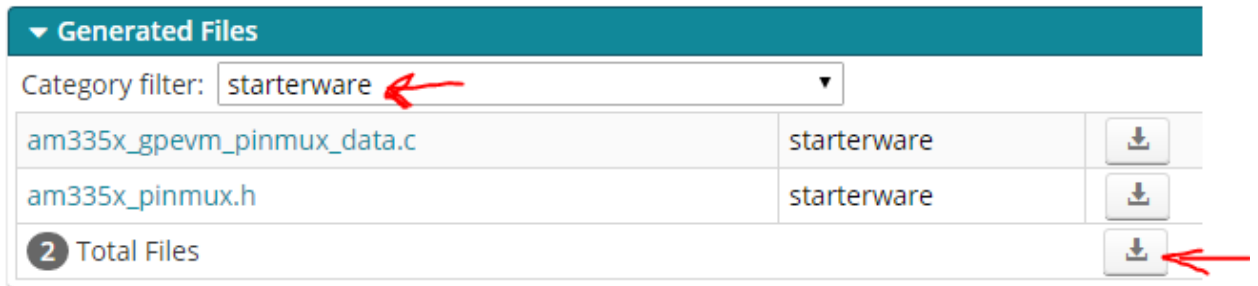
Preferred Voltage: Any Configure

IO Set: UART1_IOSet_1 View IO Sets

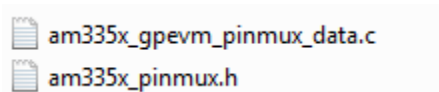
| UART Signals | UART Pins | PU | PD | Rx |
|------------------|-----------|--------------------------|--------------------------|-------------------------------------|
| ✓ rxd(uart1_rxd) | Any(D16) | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| ✓ txd(uart1_txd) | Any(D15) | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

6) Pins D15 (RX) and D16 (TX) would be utilized.

7) Change the **Category filter** to **Starterware** and download the pinmux files.



8) Once downloaded and unzipped, the following 2 files would be present.



Note the 2 below changes needed on the downloaded pinmux files:

- The current cloud based pinmux utility downloads the AM335x pinmux files with the “gpevm” in the file name. Please change `am335x_gpevm_pinmux_data.c` to `am335x_beagleboneblack_pinmux_data.c`
- Around the end of the file at `/** EVM pin configurations for EVM */` (this might change) of `am335x_beagleboneblack_pinmux_data.c`, change `gGpevmPinmuxData` to `gBbbPinmuxData`

NOTE: The above modifications apply to the current pin-mux utility. This might be fixed in a future release where the above changes might not be required, so please be aware.

- 9) These 2 files would need to be inserted into and replace the default files. In `C:\ti\pdk_am335x_1_0_0\packages\ti\starterware\board\am335x`

Example software modifications

The following areas in the AM335x Processor SDK RTOS package would need to be modified to enable UART1 on the BBB.

Clocking

- 1) At ~line 128 add a clocking instance for UART 1 in the `PRCMModuleEnable()` routine.

`C:\ti\pdk_am335x_1_0_0\packages\ti\starterware\soc\am335x\am335x_prcm.c`
 case 1:

```
enableModule(SOC_CM_PER_REGS, CM_PER_UART1_CLKCTRL,
             CM_PER_L4LS_CLKSTCTRL,
             CM_PER_L4LS_CLKSTCTRL_CLKACTIVITY_UART_GFCLK);
break;
```

- 2) `C:\ti\pdk_am335x_1_0_0\packages\ti\board\src\bbbAM335x\bbbAM335x.c` ~Line 64, add a `PRCMModuleEnable()` instance for UART1.

```
/* UART */
status = PRCMModuleEnable(CHIPDB_MOD_ID_UART, 1U, 0U);
```

UART instances

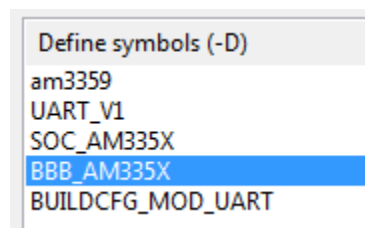
- 1) C:\ti\pdk_am335x_1_0_0\packages\ti\starterware\board\am335x\am335x_beagleboneblack.c ~line 212, specify the interrupt for UART1.

```
DEVICE_ID_CONSOLE,      /* devId */
1U,                    /* devInstNum */
CHIPDB_MOD_ID_UART,    /* ctrlModId */
0U,                   /* ctrlModInstNum */
73U, /* Interrupt Number */ /* ctrlInfo */
```

- 2) Change the UART instance in PINMUXModuleConfig() ~line 59 at
C:\ti\pdk_am335x_1_0_0\packages\ti\board\src\bbbAM335x\bbbAM335x_pinmux.c
status = PINMUXModuleConfig(CHIPDB_MOD_ID_UART, 1U, NULL);
- 3) Change the UART instance at ~line 37 of
C:\ti\pdk_am335x_1_0_0\packages\ti\board\src\bbbAM335x\include\board_cfg.h
#define BOARD_UART_INSTANCE 1
- 4) The board and Starterware libraries would need to be recompiled in order for the changes to be effective. Instructions are [here](#) on setting up the pdk environment for rebuilding.
The board library can be remade with >gmake board
Starterware can be remade with > gmake starterware (this will take a while).

CCS Project modifications

- 1) Import UART_BasicExample_evmAM335x_armExampleProject into the CCS project workspace.
- 2) Open the am335x_app_evmam335x.cfg file and ~line 49 change Board.Settings.boardName = "bbbAM335x";
- 3) In Properties>Build>GNU Compiler>Symbols change the EVM_AM335X to BBB_AM335X and click OK.

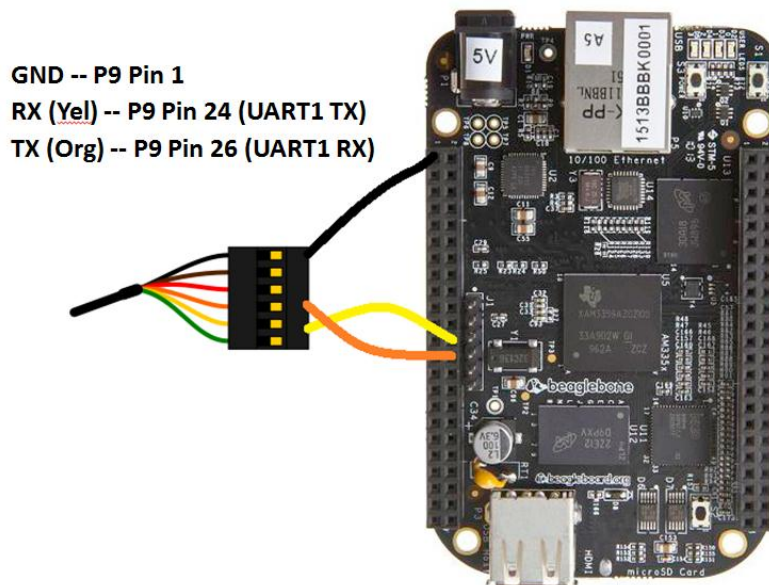


- 4) Rebuild the UART example in CCS.

Testing the changes

The default example for UART0 piped the echo characters through the FTDI cable connected to J1 on the BBB. To test UART1, please connect as shown below to pipe the UART1 data to the COM port where the FTDI cable is connected.

* The below diagram does not show the power and CCS debugger connections. It is assumed that these are also connected.



- Connect the BBB target config and load the .out file onto the target.
- Open a TeraTerm or Putty console with the COM port for the FTDI cable.
- Hit run in CCS and the following screenshot would be observed. This shows the UART example now working on UART1.

```
COM37:115200baud - Tera Term VT
File Edit Setup Control Window Help
uart driver and utils example test cases :
Enter 150 characters or press Esc
sdfsdfsdfsdf
```