

# Quick Start Guide: MicroBlaze Soft Processor Presets

## INTRODUCTION

This Quick Start Guide will walk you through creating a basic MicroBlaze™ processor system using processor preset designs. Additional resources and information can be found on the reverse side to help you tailor a MicroBlaze processor system to your exact design specifications. Features include:

- Royalty Free
- Low Power
- Highly Configurable
- Linux and RTOS support
- High Performance
- Free Development Tools

## What is the MicroBlaze Processor?

MicroBlaze is Xilinx's 32-bit RISC soft processor core, optimized for embedded applications on Xilinx devices. The MicroBlaze processor is easy to use and delivers the flexibility to select the combination of peripherals, memory, and interfaces as needed.

The MicroBlaze processor is commonly used in one of three preset configurations as shown in the table below: a simple micro-controller running bare-metal applications; a real-time processor featuring cache and a memory protection unit interfacing to tightly coupled on-chip memory running FreeRTOS; and finally, an application processor with a memory management unit running Linux. The table (below) shows performance and utilization estimates for these configurations on an Artix®-7 device.

	Microcontroller	Real-Time	Application
MHz	220	180	150
Logic Cells	1900	4000	7000
% Utilization	1%	2%	4%

\*Based on XC7A200T -3 speed grade devices

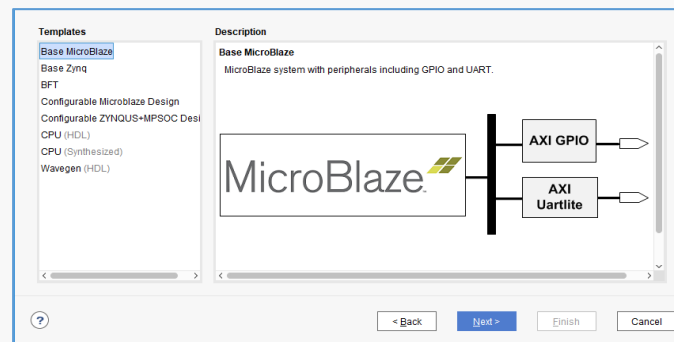
MicroBlaze can be used as a stand-alone processor in all Xilinx FPGAs or as a co-processor in a Zynq®-7000 SoC system. It can also be configured to add tamper protection and fault protection by configuring in lock-step mode as well as providing single-event upset mitigation with Triple Modular Redundancy. Designs with multiple processors can be debugged simultaneously using the Xilinx Software Development Kit (SDK).

## BEFORE YOU BEGIN

*This Quick Start Guide assumes you will be targeting an FPGA development board. If this board comes from one of Xilinx board partners, you will need to download and install a preset package. See the FAQ (next page) for links to some of our partners.*

## HARDWARE DEVELOPMENT

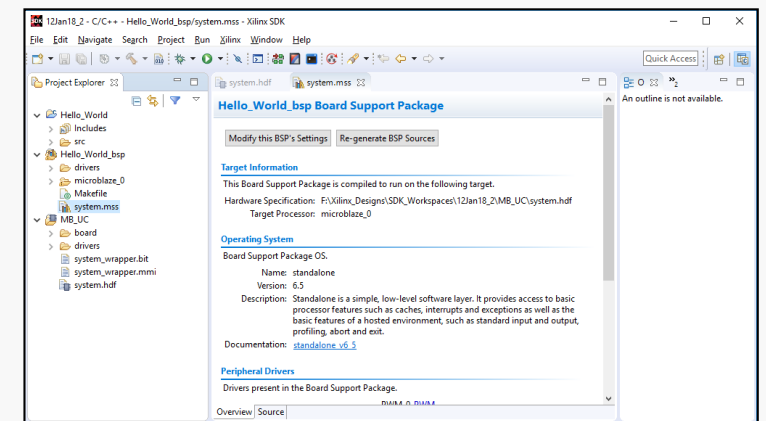
- Start Vivado® Design Suite.
- Choose **Create Project**.
- When Wizard opens, read information text and click **Next**.
- Enter a project name and location for project files (if you enter a directory that doesn't exist, it will ask if you want to create it).
- Choose **Example Project**.
- From the templates, choose one of the preset designs that works with your development board.



- Choose the target board, and click **Finish** to create the project.
- In Flow Navigator pane, click **Open Block Design** to view design.
- Click **Address Editor** tab to view peripheral addresses.
- Next, to generate the bitstream, which contains configuration data for the FPGA, choose **Generate Bitstream**.
- To launch Synthesis and Implementation runs, click **Yes**.
- When completed, click **OK** to Open Implemented Design.
- From the main toolbar, click **File** and select **Export**→**Export Hardware**. Check the box to **Include Bitstream** and click **OK**.
- To start software development with this MicroBlaze processor, select **File** → **Launch SDK** from the main toolbar. Click **OK**. SDK will open and import the hardware platform, including the MicroBlaze processor.

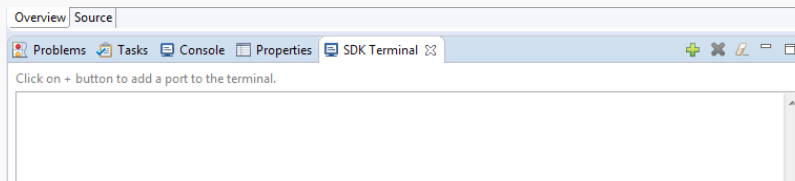
## SOFTWARE DEVELOPMENT

- Click the **New** drop-down arrow and select **Application Project**.
- Set project name to **Hello\_world** with no spaces.
- Select **Target Hardware Platform**. This is populated with the Hardware Platform built in the previous section. Alternatively, choose a pre-built Hardware Platform from the drop-down list.
- Keep all other defaults. Click **Next**.
- Select **Hello World** from the templates and click **Finish**.
- You will see three folders in **Project Explorer**:
  - **Hello\_world** contains all binaries, .C, and .H (Header) files
  - **Hello\_world\_bsp** is the board support package folder
  - **<hw\_platform>** contains information about the target hardware, which includes the memory map for the system.
- System.mss tab lists peripheral driver information and examples.

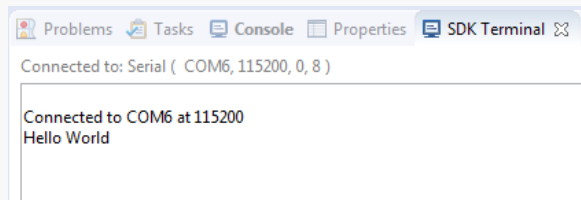


- Make sure your target board is turned on and connected to the host PC via the USB-JTAG port - this port also serves as the USB-UART connection to the MicroBlaze processor.
- On the top toolbar, click the **Program FPGA** button.
- Click **Program** to program your FPGA with your hardware design.

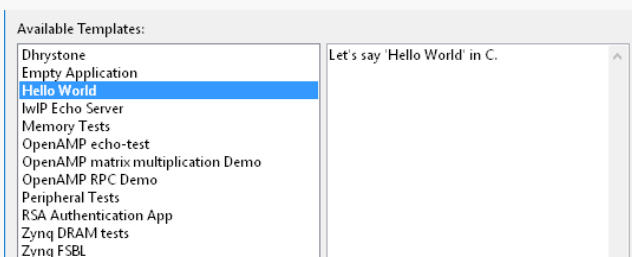
## SOFTWARE DEVELOPMENT (cont'd)



- Set up the UART terminal by clicking the **SDK Terminal** tab, then clicking the **+** to configure the correct COM Port, use settings:
  - Baud Rate: 115200
  - Data Bits: 8
  - Stop Bits: 1
  - Parity: None
  - Flow Control: None
  - Timeout(sec): [Blank]
- Now download the application by right-clicking on your **Hello\_world** project and choosing **Run As...** Select **Launch on Hardware (System Debugger)**, then click **OK**.
- Your program will run and you should see “Hello World” pop up inside of your Serial Terminal.



- Congratulations! You have created your first MicroBlaze processor application.
- Now you can try building and running other example applications, like the ones provided:



## FAQs and ADDITIONAL RESOURCES

- **How do I load third-party boards into Vivado example designs?**  
[Quick Start Instructions for Digilent ARTY A7/S7 boards](#)
- **Where should I start to learn more about the MicroBlaze processor?**  
Visit the [MicroBlaze Design Hub](#). It contains links to documentation, wikis, and video tutorials that provide lots of information. Most of the document links in this FAQ can also be found there.
- **Where can I find specific details about the MicroBlaze processor?**  
Go to: [UG984 - MicroBlaze Processor Reference Guide](#).
- **How can I estimate the size and performance for my configured MicroBlaze processor?**  
Go to: [MicroBlaze Performance Metrics](#) as a starting point.
- **Where can I find a more comprehensive tutorial?**  
Go to: [UG940 - Lab 3: Using the Embedded MicroBlaze Processor](#).
- **Where do I go for more details about creating a Vivado tool design?**  
Visit our [Vivado Design Hubs](#).
- **Do I need to be in the Vivado tools to start SDK?**  
No. SDK is an independent tool that can be started from the Xilinx Design Tools. However, you'll need a MicroBlaze processor hardware design file (.hdf) file to target for your software development.
- **What do I do if the board I'm targeting is not on the list?**  
Many board vendors provide board files that can be added to Vivado and SDK. Contact the specific manufacturer for these files.  
[Digilent Reference Guide for ARTY S7](#)  
[Avnet Reference Designs & Tutorials for MiniZed](#)
- **What if I need to make changes to my hardware design?**  
Close the SDK window and make the required HW design edits in the Vivado tools, then follow the sequence for bit file generation. This updated hardware design must then be exported from the Vivado tools and imported into SDK.
- **How do I expand the capability of my evaluation board?**  
[PMODs](#), [Arduino shields](#), and [FMC cards](#) can be used to expand the capabilities of our evaluation boards.
- **How do I create a bootable flash image that contains my bitstream and application?**  
See chapter 7 of [UG898](#). In Vivado, **Tools** → **Associate ELF** files. In SDK, **Xilinx Tools** → **Program FPGA** (select ELF for MicroBlaze).

- **What happens when I export hardware and launch SDK?**  
A file called system.hdf is created. This hardware definition file contains HW specifications, IP interfaces, external signal information, and local memory address information. This is used by SDK for application development and debug.
- **How do I communicate between Zynq-7000 SoC and MicroBlaze?**  
See this QTV on YouTube: [Zynq and MicroBlaze IOP Block, OCM, and Memory Resource Sharing](#).
- **How do I debug multiple processors in a single system?**  
[YouTube](#) [Heterogeneous Multicore Debugging with Xilinx SDK](#).
- **How much FPGA memory can the MicroBlaze processor access?**  
MicroBlaze systems can be created that access all the available memory on an FPGA. But this comes at the cost of lower F<sub>MAX</sub>. Typical MicroBlaze implementations use 128KB or less.
- **What OS & libraries are supported in SDK for MicroBlaze?**  
See [Supported Operating Systems](#) & [UG643 - OS and Libraries Guide](#).
- **Can I run Linux or other OSs on the MicroBlaze processor?**  
Yes. For best performance with Linux, you need to configure the MicroBlaze processor like a standard applications processor.
- **How do I create a Linux bootloader for the MicroBlaze processor?**  
Go to: [Build U-Boot for MicroBlaze](#).

## Resources

- [MicroBlaze Documentation Design Hub](#)
- [MicroBlaze Getting Started Wiki](#)
- [MicroBlaze Soft Processor Core Product Page](#)
- [Using the MicroBlaze Processor to Accelerate Cost-Sensitive Embedded System Development](#)
- [Document Navigator Embedded Hub](#)
- [Vivado Tutorials](#)
- [Xilinx SDK Tools Help](#)
- [Knowledge Base Answer Records](#)
- [Zedboard.org](#)
- [Digilent.com](#)