



MICROCHIP VectorBlox Video Kit Demo User Guide

[Home](#) » [MICROCHIP](#) » MICROCHIP VectorBlox Video Kit Demo User Guide

MICROCHIP VectorBlox Video Kit Demo User Guide



Contents [[hide](#)

- 1 Introduction
- 2 Purpose
 - 2.1 Intended Audience
- 3 References
 - 3.1 Known Issue
- 4 Design Requirements
- 5 Development Kit for Demo
- 6 Demo Design Description
 - 6.1 System Design
- 7 Setting Up the Demo
 - 7.1 Setting Up the Hardware
 - 7.2 Programming the Polar Fire Video Kit
 - 7.3 Programming the Device
 - 7.4 Running the Demo
- 8 Running Alternate Models
 - 8.1 Obtaining Model File
 - 8.2 Modifying the SPI Flash Configuration
 - 8.3 Soft Console Project
 - 8.4 Running the Mi-V Program
- 9 Revision History
- 10 The Microchip Website
- 11 Product Change Notification Service
- 12 Microchip Devices Code Protection Feature
 - 12.1 Legal Notice
- 13 Trademarks
- 14 Quality Management System
- 15 Customer Support
 - 15.1 Worldwide Sales and Service
- 16 Documents / Resources
- 17 Related Posts

Introduction

This document describes how to run the Core Vector Blox Neural Network using the Polar Fire Video Kit, the Dual Camera sensor module, an HDMI monitor, and optionally a HDMI source such as a laptop. The demo design features a fully integrated solution developed using Microchip Libero® SoC software to help customers evaluate PolarFire FPGA in Neural Network Vision applications and to build prototypes quickly. For more information, see Smart Embedded Vision.

The demo demonstrates the following functions:

- MIPI CSI-2 RX to read one of the cameras
- HDMI display controller
- VectorBlox CNN acceleration of Tiny YOLOv3
- VectorBlox CNN acceleration of MobileNet V1
- Vectorblox CNN acceleration of a face recognition solution featuring Retinaface Mobilenet and Arcface
- Image enhancements such as contrast, brightness, and color balance

The PolarFire Video Kit (MPF300-VIDEO-KIT-NS) includes the following components:

- A 300K LE FPGA (MPF300T, FCG1152)
- HDMI 1.4 transmitter (ADV7511) chipset and corresponding connector
- HDMI 2.0 with rail clamps, ReDrivers, and corresponding connectors
- Dual camera sensor featuring IMX334 Sony image sensor
- Image sensor interface to support up to two MIPI CSI-2 cameras
- Display Serial Interface (DSI)
- NVIDIA® Jetson interface (MIPI CSI-2 TX connector)
- A High Pin Count (HPC) FMC connector to connect to high-speed interfaces (such as 12G-SDI and USXGMII)

For more information about the video kit, see Polar Fire FPGA Video and Imaging Kit

Purpose

This demo is for Core Vector Blox neural network acceleration on the Polar Fire® field-programmable gate array (FPGA) devices. This document provides instructions on how to use the corresponding reference design.

Intended Audience

This demo guide is intended for:

- FPGA designers
- Firmware designers
- System level designers
- Data scientists

References

The following documents are referred in this demo guide.

- CoreVectorBlox SDK Programmer's Guide
- CoreVectorBlox IP Handbook

Known Issue

There is a known Reset bug where sometimes the frame buffer management comes out of Reset incorrectly. If the output video is out of sync, try power cycling the board.

Design Requirements

The following table lists the hardware and software required to run the demo.

Design Requirements	Description
Hardware Requirements	
Polar Fire Video Kit Development Board	MPF300-VIDEO-KIT-NS
USB A to mini-B cable(1)	<p>Required for the following:</p> <ul style="list-style-type: none"> • FPGA programming and SPI Flash programming • Running the modified Mi-V C code from Soft Console
HDMI cable(1)	HDMI A Male to Male cable
Power adapter(1)	12V, 5A
HDMI monitor	A 1920 x1080 60 Hz resolution monitor for the HDMI 1.4 TX port
Host PC	A host PC with a USB port and HDMI output
Software Requirements	
Libero® System-on-Chip (SoC) v2021.2	<p>You must install the full Libero SoC software and not just the programming tools to program the SPI Flash, which cannot be done from FP Express.</p> <p>A Libero license is necessary; the video kit comes with a Gold license or an evaluation license that can be obtained from the Licensing tab of the following page.</p> <p>Libero SoC v12.0 and later</p>
Core Vectorblox License	<p>To configure and synthesize the Core Vector blox IP, a license is required. It is available at SoC portal.</p>

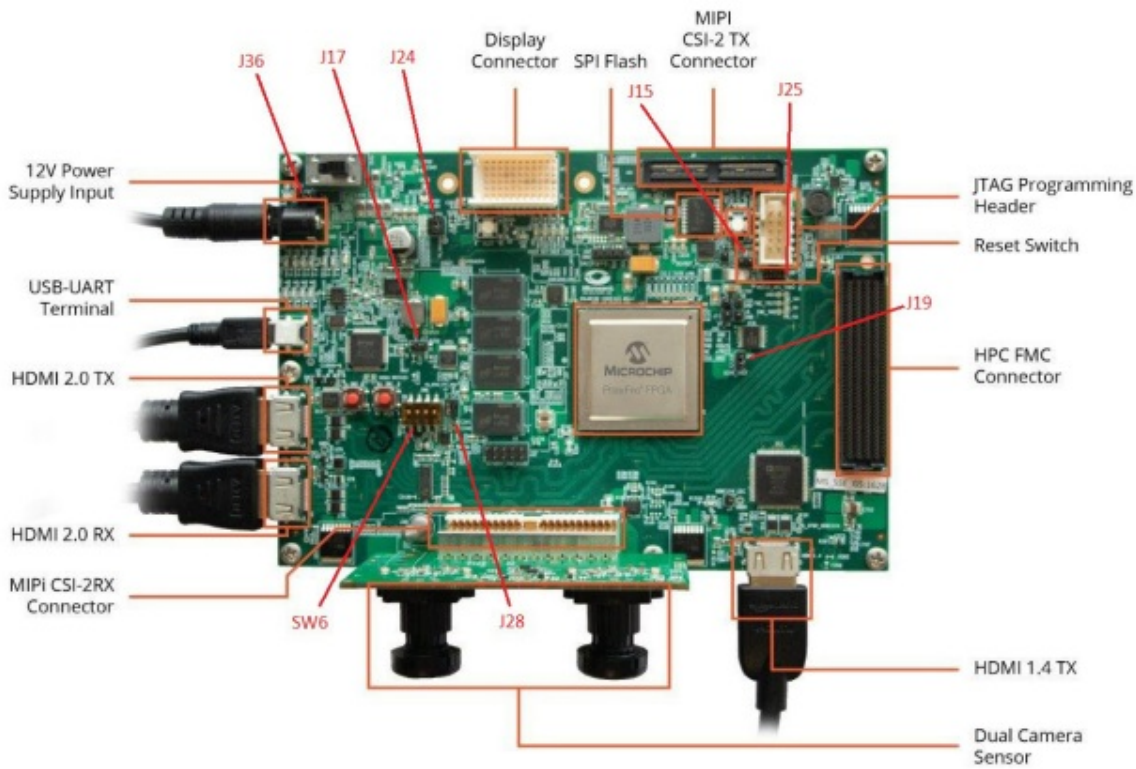
Note:

1. Included with the Polar Fire Video Kit.

Development Kit for Demo

The following figure highlights the features of Polar Fire Video Kit.

Figure 3-1. Polar Fire Video Kit Features.



The following table provides the jumper position and functionality for the jumper settings.

Table 3-1. Jumper Description

Jumper	Default Position	Functionality
J15	Open	SPI Slave and Master mode selection. Default: SPI master.
J17	Open	100K PD for TRSTn. Default: 1K PD is connected.
J19	Pin1 and 2	Default: XCVR_VREF is connected to GND.
J28	Pin 1 and 2	Default: Programming through the FTDI.
J24	Pin 2 and 4	Default: VDDAUX4 voltage is set to 3V3.
J25	Pin 5 and 6	Default: Bank4 voltage is set to 1V8.
J36	Pin 1 and 2	Default: Board power-up through SW4.
SW6	ON	Chooses between the face recognition demo and the mobilenet/yolo demo.

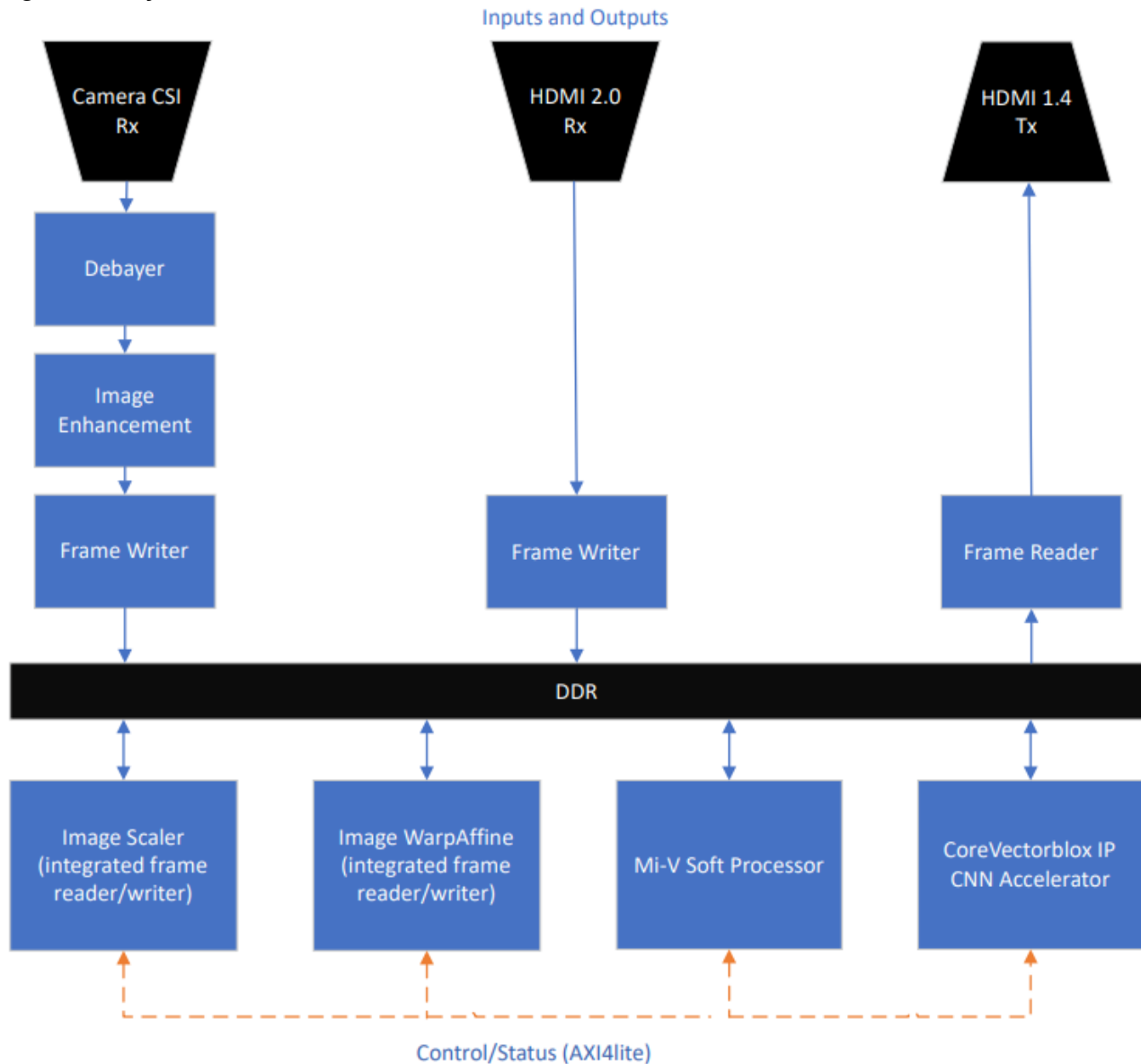
Demo Design Description

The following section provides an overview of the dataflow in the demo design. This system design allows the mobilenet/yolo demo and the face recognition demo to run on the same hardware.

System Design

The following diagram illustrates an overview of the dataflow in the design.

Figure 4-1. System Dataflow



Sequence of data flow shown in the figure above is as follows:

1. Input

- **a.** Received from MIPI CSI 2: Video frame data is passed through a debayer and Image Enhancement block to a Frame Writer which writes the video data to DDR.
- **b.** Received from HDMI Rx: If a source is connected to J35, the video frame data is sent directly to a Frame Writer which writes the video data to DDR.

2. Based on demo requirements, Mi-V instructs the Image Scaler and/or the Image WarpAffine block to process the input frame appropriately for CNN(s). If an HDMI source is present, the input frame is taken from HDMI,

otherwise it is taken from the camera. The result of scaling and warping is written to DDR.

3. Mi-V instructs CoreVectorBlox IP to run the appropriate CNN using the scaled and warped image as input to CNN. The result of CNN is written back to DDR by CoreVectorBlox IP.
4. Mi-V reads the result of CNN from DDR and runs post-processing software routines. The result is then drawn on the original Input Frame Buffer.
5. The Frame Reader reads the frame and streams the frame data to the HDMI Tx block.

Setting Up the Demo

The following steps describe how to setup the demo.

1. Setting up the Hardware
2. Programming the Polar Fire Device
3. Programming the SPI Flash

Setting Up the Hardware

Setting up the hardware involves interfacing the dual camera sensor module and the HDMI monitor with the Polar Fire Video Kit, and verifying the jumper settings.

Perform the following steps.

1. Connect the J1 connector of the dual camera sensor module to the J5 interface of the video kit. The video kit is already shipped with this.
2. Connect the Full HD HDMI (1080P) monitor to J2 (HDMI 1.4 TX port) of the video kit using the HDMI cable.
3. Connect the host PC to J12 of the video kit using the USB mini cable.
4. Connect the power supply cable to J20 of the video kit.
5. Ensure that the jumper settings are set on the video kit. The video kit is shipped in this configuration. For jumper position and functionality, see Table 3-1.
6. Power-up the HDMI monitor.
7. Power-up the board using the SW4 slide switch.
8. Optionally, if running the face recognition demo, connect HDMI source playing SampleVideo.mp4 (supplied with the project ZIP file) full screen to the HDMI 2.0 RX port (J35).

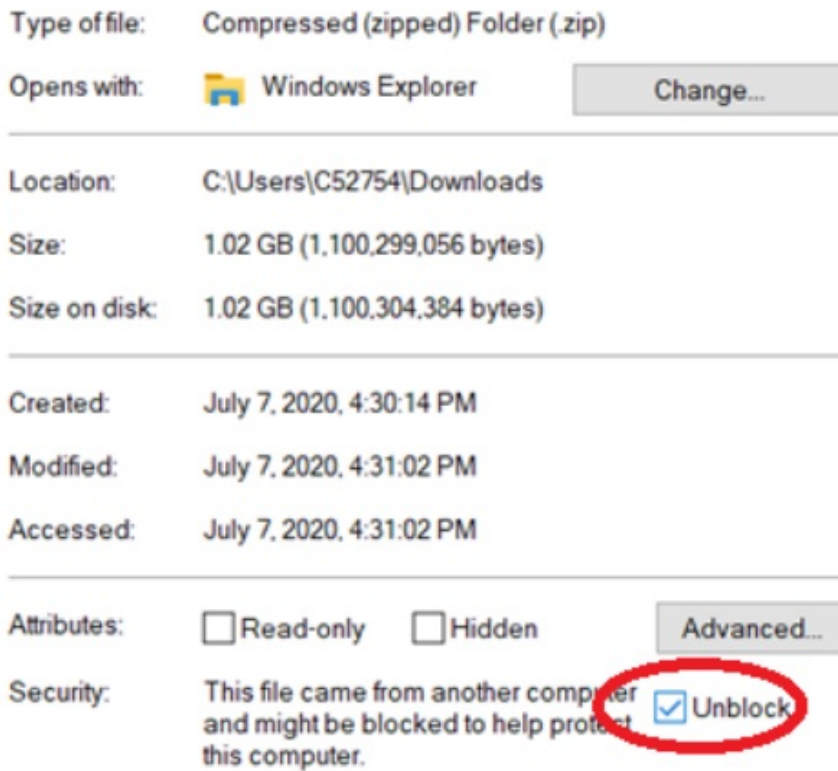
The Polar Fire dual camera video and imaging hardware setup is completed.

Programming the Polar Fire Video Kit

The following section describes how to program the Polar Fire device and run the demo.

Extracting the Source files and Opening the Project

Before unzipping the archive containing the libero project, first “unblock” the file. This is necessary to ensure that Windows does not change the timestamps of the files during extraction. To unblock the file, right click the zip file, select Properties and check Unblock, then click OK.



After unzipping the archive, launch Libero SoC v2021.2, and open the vectorblox_videokit_v1.2/vectorblox_videokit_v1.2.prjx project file.

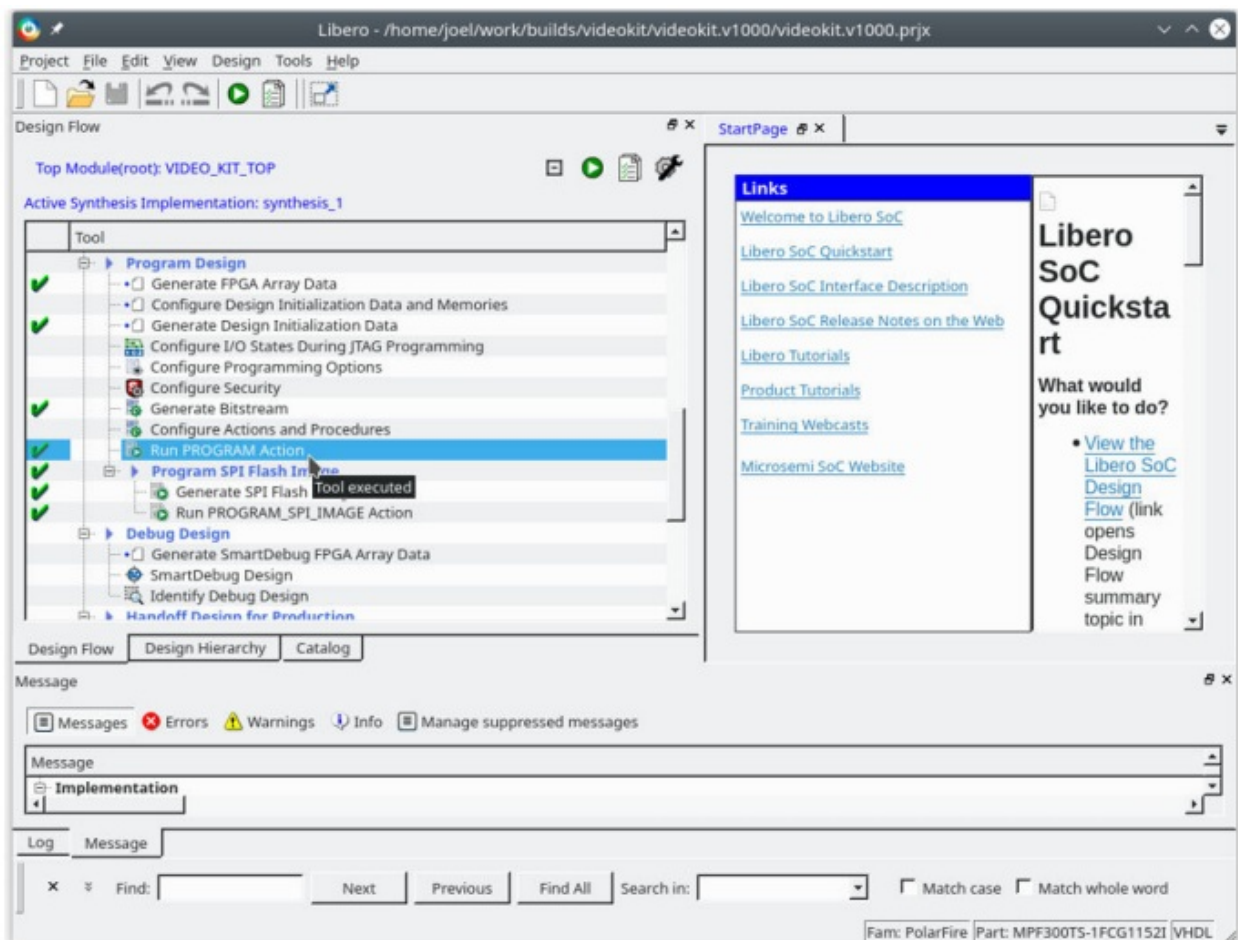
Note: You may be prompted to update Libero SoC or individual IP cores, ignore these prompts.

Programming the Device

Perform the following steps.

1. In the Design Flow window, double click Run PROGRAM Action

Figure 5-1. Run PROGRAM ACTION

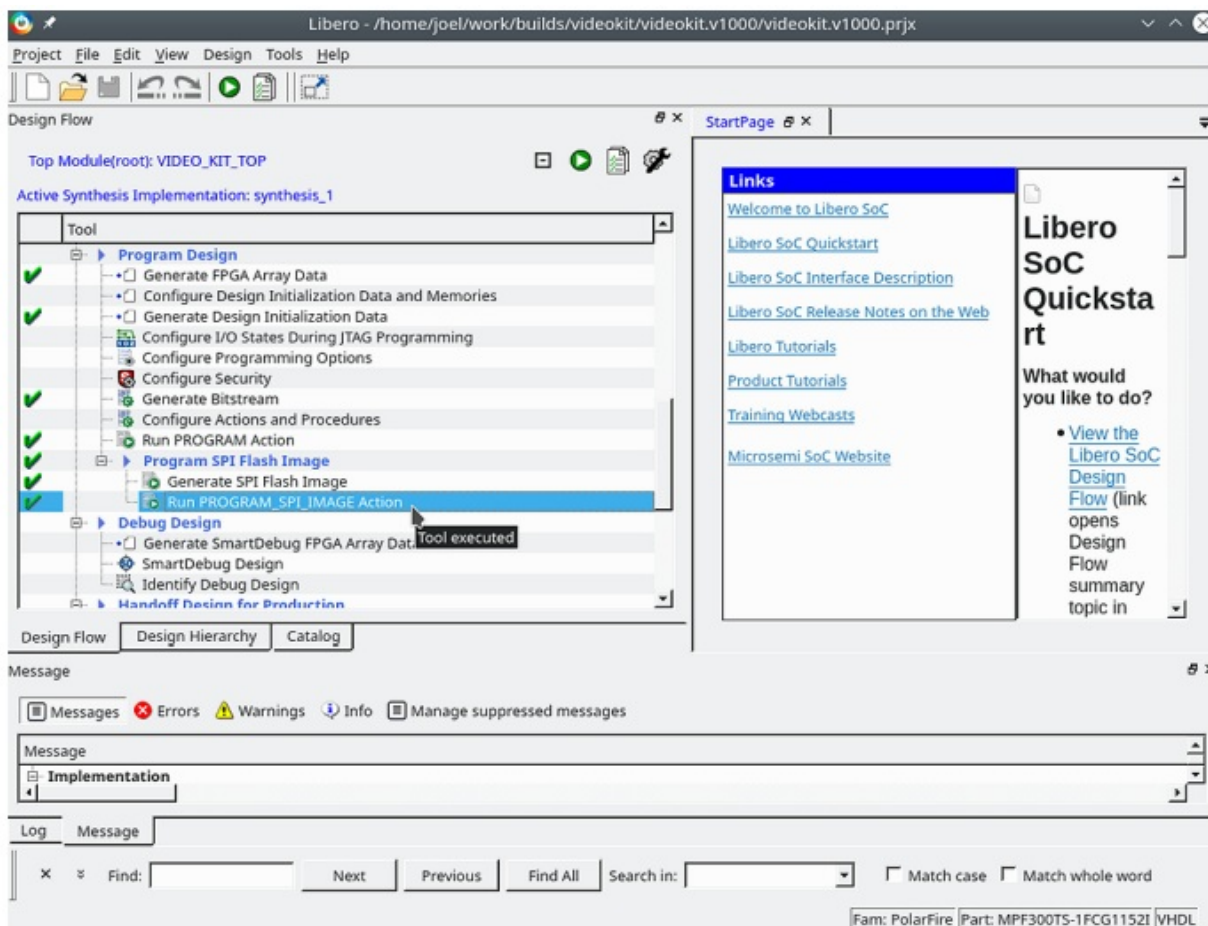


2. Double click Run PROGRAM_SPI_IMAGE Action and wait. This will take some time.

Notes:

- In Windows, you might be prompted with a firewall popup.
- Ignore any warnings about misaligned sectors.

Figure 5-2. Run PROGRAM_SPI_IMAGE Action.



Running the Demo

Power cycle the board with SW4 to start the demo.

Note: Sometimes the demo might appear to start correctly without a power cycle but actually be in an Incoherent state.

The startup takes a few minutes. The following events occur during the startup: the camera is calibrated to the brightness of the environment, the firmware and models are read from the Flash into DDR

After the startup is completed, one of the following occurs:

- With SW6.1 set to “ON”, the demo will switch every 5 seconds between Mobile Net V1 and TinyYoloV3.
- With SW6.1 set to “OFF”, the demo will run Retina face Mobile net to find faces in the frame.

Note: The device (for example, laptop) should be running the mp4 video connected over HDMI to J35 for this.

If faces are found, it will run Arc face for one of the faces to identify the face. If more than one face is found, the other faces will rely on the Kalman filter tracking to keep track of the name of the face.

For face recognition and mobilenet demos, you can select the source of the frames that needs to be processed by the network. The source is determined by connecting to the J35 connector. An HDMI cable connecting a video source (for example, from a laptop) will take priority over the image sensor coming with the video kit. If the HDMI cable is not connected to J35, the image sensor will be selected as the source.

Running Alternate Models

When SW6.1 is set to “ON”, the project that is provided here runs only two models: MobileNet V1 and Tiny YOLOv3.

However, it is capable of running many other networks. In this document, you will see an example of swapping out Tiny YOLOv3 for Tiny YOLOv2

In this demo, four models are required in the specified order:

- A face detection model
- A face recognition model
- Two classification/object-detection models

The first two are used in the Face Recognition mode, and the latter two are cycled between in the Classification/Object-Detection mode.

Currently, post-processing routines exist for TinyYoloV2 (VOC), TinyYoloV3 (COCO), Retinaface, Blazeface, Sphereface/Arcface (face recognition), and Imagenet networks. Users will have to write additional post-processing code to run other networks.

The following sections describe how to run the alternate models

Obtaining Model File

The model files can be obtained by running the tutorial available in the VectorBlox SDK in Github. Instructions for running the tutorials can be found in the Programmer's Guide available as part of the SDK documentation. The artifact generated from the tutorial that needs to be stored is yolov2-tiny-voc.hex. This hex file will be added to the SPI Flash on the board.

Modifying the SPI Flash Configuration

Perform the following steps in Libero.

1. Invoke the "Configure Design Initialization Data and Memories" tool in the Libero Design Flow.
2. To add a new model, click the Add button, and select the Add Data Storage Client option. Change the path to point to yolov2-tiny-voc.hex (file described in the proceeding section, Obtaining Model File).
Note: Ensure the address range does not overlap with other clients in the Flash memory.
3. Click Apply.
4. In the Design Flow window, double click Run PROGRAM_SPI_FLASH Action (see Step 3 in the Programming the Device section).
5. After the SPI programming is complete, power cycle the board using SW4.

Figure 6-1. SPI Configuration after Modification.

Design Initialization

uPROM

sNVM

SPI Flash

Fabric RAMs

Apply

Discard

Help

☐ Enable Auto Update


SPI Flash memory size: 131,072 KB

Usage statistics

Available memory (KB): 131071

Used memory (KB): 70433

Free memory (KB): 60638



SPI Flash Clients

Add...

Edit...

Delete

Program All

Bypass All

Program	Name	Type	Index	STAGE3 Init Client	Start Address	End Address	Content	Design Version	User Security	Bypass Back Level Protection
<input checked="" type="checkbox"/>	ascii_characters	Data Storage			0x40400	0xc1dba7f	hex\ascii_characters.hex		N/A	N/A
<input checked="" type="checkbox"/>	firmware	Data Storage			0xc1dba80	0xc3dba7f	hex\firmware.hex		N/A	N/A
<input checked="" type="checkbox"/>	yolov3-tiny	Data Storage			0xc3dba80	0xc1614407	hex\yolov3-tiny.hex		N/A	N/A
<input checked="" type="checkbox"/>	mobilenet-v1-1.0-224	Data Storage			0xc1614408	0xc224974f	hex\mobilenet-v1-1.0-224.hex		N/A	N/A
<input checked="" type="checkbox"/>	retinaface.mobilenet	Data Storage			0xc2249750	0xc23f9c67	hex\retinaface.mobilenet.hex		N/A	N/A
<input checked="" type="checkbox"/>	mobilefacenet-arcface	Data Storage			0xc23f9c68	0xc26017a7	hex\mobilefacenet-arcface.hex		N/A	N/A
<input checked="" type="checkbox"/>	INIT_STAGE3_SPI_CLIENT	Design Initialization			0x400	0x2e6ff	designer\VIDEO_KIT_TOP\VIDEO_KIT_TOP_uic.bin		N/A	N/A
<input checked="" type="checkbox"/>	yolov2tinyvoc	Data Storage			0xc26017a8	0x44da727	..\yolov2-tiny-voc.hex		N/A	N/A

Soft Console Project

Before the new model is run on the FPGA, the software running on Mi-V must be modified as described in this section.

A SoftConsole project is located in the Libero Design zip archive at Download Directory/vectorblox_videokit_v1.1/soft console. Open that directory as a workspace with Soft Console 6.2 or later. In the VideoKit project locate and open main.c.

The following code can be seen on line 187.

```
struct model_descr_t models[] = {
    {"Retinaface", 0x2249750, 256, RETINAFACE, 30},
    {"Arcface", 0x23f9c68, 112, ARCFACE, 30},
    {"MobileNet V1", 0x1614408, 224, IMAGENET, 10},
    {"Tiny Yolo V3 COCO", 0x3dba80, 416, TINYYOLOV3, 30},
};
```

Soft Console 6.2 or later

```
struct model_descr_t models[] = {
    {"Retinaface", 0x2249750, 256, RETINAFACE, 30},
    {"Arcface", 0x23f9c68, 112, ARCFACE, 30},
    {"MobileNet V1", 0x1614408, 224, IMAGENET, 10},
    {"Tiny Yolo V2 VOC", 0x26017a8, 416, TINYYOLOV2, 30},
};
```

Where, parameters in the structure are as follows:

- Display name of the model
- Address in the SPI Memory in which the model is stored.
- Resolution of the square input image needed for the network.

Note: Resolution of the network should be documented by the network provider. For instance, Open VINO's open-model-zoo; it can link to https://github.com/openvinotoolkit/open_model_zoo.

- The type of postprocessing for displaying the network. Currently, IMAGENET (Resnet/Mobilenet/etc.), Retinaface, BlazeFace, SpheroFace/ArcFace (face recognition), TinyYoloV2, and TinyYoloV3 are implemented.
- The maximum number of frames per second to display. Used to reduce the rate of change of displayed labels to make the network output readable.

Ensure you have the four models listed for full demo functionality, in the following order:

- A face detection model
- A face recognition model
- Two classification/object-detection models

After these modifications are performed, save the main.c file, and build the project. The software is now ready to run and models can be executed.

Running the Mi-V Program

Make sure to use FP5 and jumper J28 connects Pin 1 to Pin 2. To run the Mi-V program, click the button in the toolbar.

Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Revision	Date	Description
D	09/2021	Updated to 1.2 Release
C	05/2021	Updated to 1.1 Release
B	11/2020	Updated to 1.0 Release
A	08/2020	Initial Revision

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- Product Support – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- General Technical Support – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- Business of Microchip – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE. IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP’S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, Any Rate, AVR, AVR logo, AVR Freaks, Bes Time, Bit Cloud, Crypto Memory, Crypto RF, dsPIC, flex PWR, HELDO, IGLOO, Juke Blox, Kee Loq, Kleer, LAN Check, Link MD, maX Stylus, maX Touch, Media LB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, Opto Lyzer, PIC, pico Power, PICSTART, PIC32 logo, Polar Fire, Prochip Designer, Q Touch, SAM-BA, Sen Genuity, Spy NIC, SST, SST Logo, Super Flash, Symmetricom, Sync Server, Tachyon, Time Source, tiny AVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Agile Switch, APT, Clock Works, The Embedded Control Solutions Company, Ether Synch, Flash tec, Hyper Speed Control, Hyper Light Load, Intel liMOS, Libero, motor Bench, m Touch, Power mite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet- Wire, Smart Fusion, Sync World, Temux, Time Cesium, Time Hub, Time Pictra, Time Provider, True Time, Win Path, and ZL are registered trademarks of Microchip Technology Incorporated in the

U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, Espresso T1S, EtherGREEN, GridTime, IdealBridge, In-Circuit Serial Programming, ICSP, INICnet, Intelligent Paralleling, Inter-Chip Connectivity, JitterBlocker, Knob-on-Display, maxCrypto, maxView, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, NVM Express, NVMe, Omniscent Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, RTAX, RTG4, SAMICE, Serial Quad I/O, simpleMAP, SimpliPHY, SmartBuffer, SmartHLS, SMART-I.S., storClad, SQL, SuperSwitcher, SuperSwitcher II, Switchtec, SynchroPHY, Total Endurance, TSHARC, USBCheck, VariSense, VectorBlox, VeriPHY, ViewSpan, WiperLock, XpressConnect, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Worldwide Sales and Service

Corporate Office

2355 West Chandler Blvd.
Chandler, AZ 85224-6199

Tel: 480-792-7200

Fax: 480-792-7277


Technical Support:

www.microchip.com/support

Web Address:

www.microchip.com



	<p>MICROCHIP VectorBlox Video Kit Demo [pdf] User Guide</p> <p>VectorBlox Video Kit Demo, Video Kit Demo, VectorBlox Video Kit, Video Kit</p>
---	---