

CON3, SSR, MAXSONAR PMOD IP Tutorial

Team 5

Table of Contents

1. Introduction	3
PMOD Connection Guidelines	3
Project Setup for the Hub Board	3
Block Diagram for the Hub Board	4
Description of Testing File	4
2. servo_con3	6
Source	6
Initial Setup	6
Description	6
3. ssrelay	7
Source	7
Initial Setup	7
Description	7
4. MAXSONAR	8
Source	8
Initial Setup	8
Description	8

1. Introduction

This tutorial describes the configuration, usage, and APIs for the following PMOD IPs targeting the Nexys Video Board:

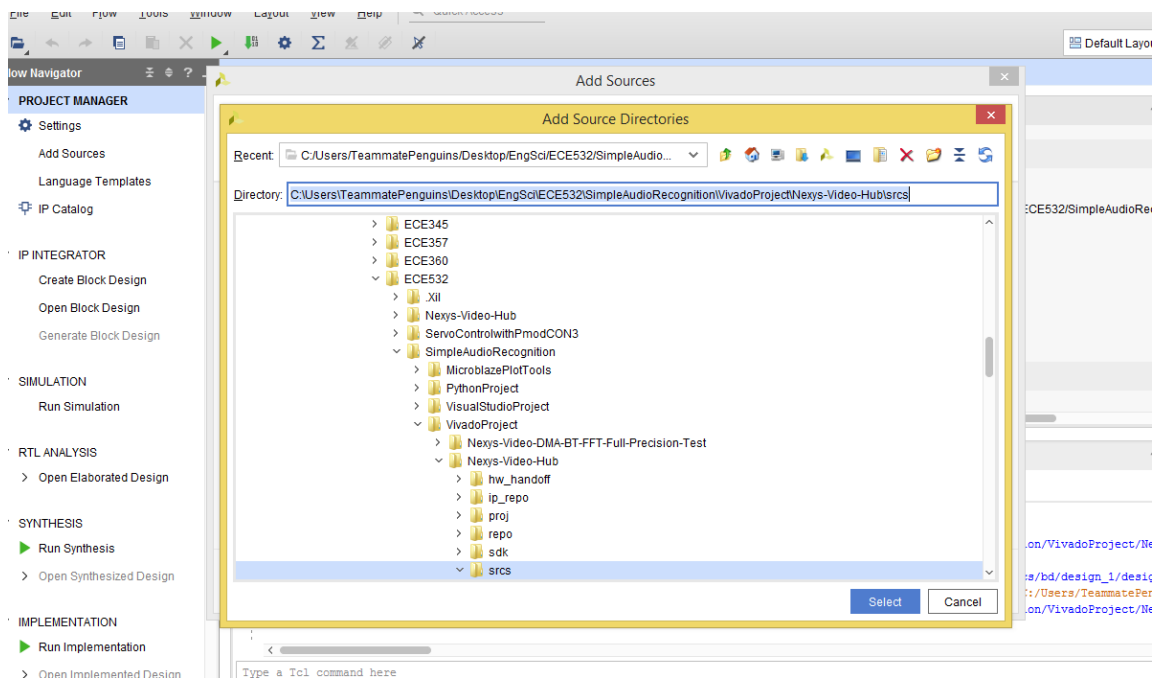
- servo_con3: custom IP for servo controller PMOD CON3
- ssrelay: custom IP for relay switch PMOD SSR
- PmodMAXSONAR: Digilent IP for distance detection
- PmodBT2: Digilent IP for Bluetooth communication (see https://github.com/CaptainPenguins/SimpleAudioRecognition/blob/master/doc/Floating_point_DMA_FFT_BT_Tutorial.pdf for details)

PMOD Connection Guidelines

PMOD	Pins on Board
CON3	JXADC[7:10]
SSR	JA[1:4]
MAXSONAR	JC[1:4]
BT2	JB

Project Setup for the Hub Board

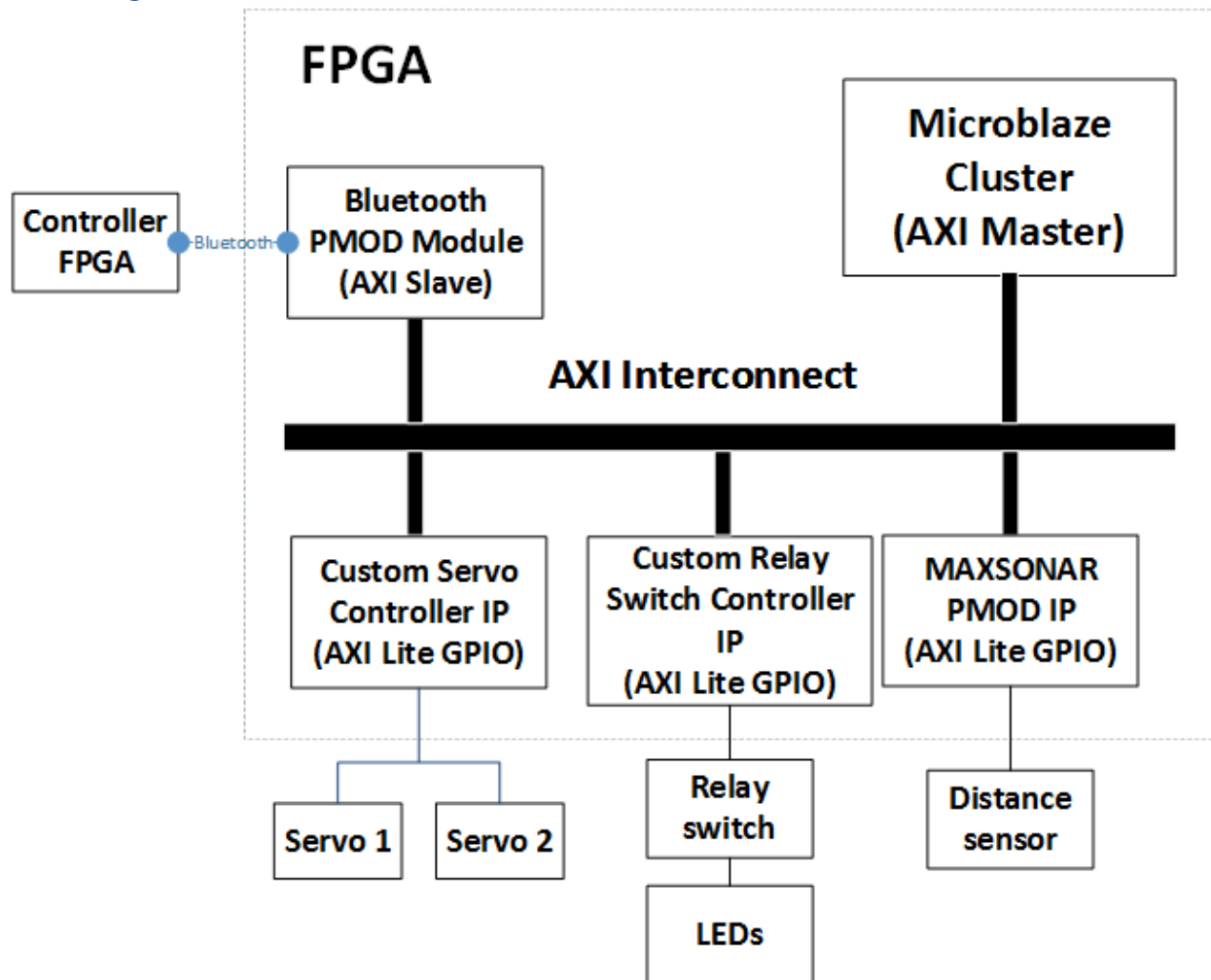
1. To start, download the project file at:
<https://github.com/CaptainPenguins/SimpleAudioRecognition>
2. Place the project file in a directory path with no spaces
3. Open Vivaldo, cd to `.../SimpleAudioRecognition/VivadoProject/Nexys-Video-Hub/proj` directory
4. Type `source create_project.tcl`
5. Add all source files, Add Sources->Add or create design sources->Next->Add Directories->srcs



6. Add the path for the following two IP repository
 - 1) Digilent IPs (download from <https://github.com/Digilent/vivado-library/releases>)
 - 2) The ip_repo folder inside the project

7. Validate the block design.
8. Create HDL wrapper of the block design and generate bit-stream
9. Export hardware and launch SDK
10. In the SDK window, go to File -> Import Project from File System
11. Import the two project folders inside the sdk folder: *hub_fpga* and *hub_fpga_bsp*
12. Program the FPGA
13. Run GDB debugger configuration with outputs to console

Block Diagram for the Hub Board



Description of Testing File

The testing file for all the PMOD IPs (both custom and Digilent) can be found at ...\\SimpleAudioRecognition\\VivadoProject\\Nexys-Video-Hub\\sdk\\hub_fpga\\src\\hub_control.c. This file contains the following seven different functions for controlling the PMODs:

- **void DoorOpen(volatile unsigned int*):** continuously writing to the address of the targeted servo motor to open the door. The parameters are pre-set, but they can be modified to adjust the position and turning speed.
- **void DoorClose(volatile unsigned int*):** continuously writing to the address of the targeted servo motor to open the door. The parameters are pre-set, but they can be modified to adjust the position and turning speed.
- **void LockUp(volatile unsigned int*):** continuously writing to the address of the targeted servo motor to rotate the lock to open position. The parameters are pre-set, but they can be modified to adjust the position and turning speed.
- **void LockDown(volatile unsigned int*):** continuously writing to the address of the targeted servo motor to rotate the lock to close position. The parameters are pre-set, but they can be modified to adjust the position and turning speed.
- **void LightOn(volatile unsigned int*):** writing 0x1 to the address of the relay switch
- **void LightOff(volatile unsigned int*):** writing 0x1 to the address of the relay switch
- **void Enter(PmodMAXSONAR mySonar, volatile unsigned int* ssr_ptr, volatile unsigned int* servo_con_ptr):** continuously polling the reading from the MAXSONAR PMOD. If the reading stays within 6-8 inches for about 2.5 seconds (determined by the constant THRESHOLD), the call the functions LightOn, LockUp, OpenDoor in order. Then, return to polling from the MAXSONAR PMOD. If the reading stays out of the range 6-8 inches for about 2.5 seconds (determined by the constant THRESHOLD), the call the functions LightOff, DoorClose, LockDown in order

The following functions will be called when special characters are received from the Bluetooth channel:

- 'A': LockUp ->DoorOpen
- 'B': DoorClose->LockDown
- 'C': LightOn
- 'D': LightOff
- 'E': Enter

Debugging print statements are already setup at appropriate places for the ease of seeing what data are written into the addresses of various PMODs.

2. servo_con3

Digilent does not provide an IP for the CON3 servo controller PMOD. This is a custom IP built from a sample Verilog project into an AXI Lite Slave

Source

User guide: https://reference.digilentinc.com/pmod/pmod/con3/user_guide

Verilog project: <http://www.instructables.com/id/Controlling-Servos-on-FPGA/>

Custom IP directory: ...*SimpleAudioRecognition*\VivadoProject\Nexys-Video-Hub\ip_repo\servo_con3_1.0

Initial Setup

1. Connect the CON3 PMOD to **JXADC[7:10]**.
2. Connect slot 1 to the servo motor that controls the door.
3. Connect slot 2 to the servo motor that rotates the lock.
4. Power the CON3 PMOD module with a 5 V external battery source.

Description

The servo_con3 IP has an AXI Lite interface, in which two instances of Servo_interface modules are instantiated. This module takes a 32-bit input and converts it to a 9-bit value, the desired servo angle in degrees, using the module sw_to_angle. Then, it instantiates the module angle_decoder that converts the angle to a 20-bit value, representing the PWM duration, that depends on the model of the servo motor. A counter and a comparator is used to output a 1-bit PWM signal that is high if the counter is within the duration, and low otherwise. The refresh cycle is set to be 100 MHz in the module counter.

3. ssrelay

Digilent does not provide an IP for the CON3 servo controller PMOD. This is a custom IP built from scratch into an AXI Lite Slave

Source

User guide: <https://reference.digilentinc.com/reference/pmod/pmodssr/start?redirect=1>

Custom IP directory: ...*SimpleAudioRecognition*\VivadoProject\Nexys-Video-Hub\ip_repo\ssrelay_1.0

Initial Setup

1. Connect the SSR PMOD to **JA[1:4]**.
2. Connect a battery and a few LEDs in series with the PMOD

Description

The ssrelay IP has an AXI Lite interface, in which an instance of the module top is created. This module takes a 32-bit value as input, but only cares about the least significant bit. This bit is sent to the PMOD through the 6-pin interface. A built-in LED turns on when the signal is high and turns off otherwise. This PMOD is just a switch. Therefore, an appropriate battery pack and actual light bulbs/LEDs are required.

4. MAXSONAR

This is a packaged IP provided by Diligent. It is contained in the Diligent repository mentioned on page 3.

Source

User guide: <https://reference.digilentinc.com/reference/pmod/pmodmaxsonar/start?redirect=1>

Initial Setup

1. Connect the SSR PMOD to **JC[1:4]**. Might need an extension cable for good results and the option of placing the PMOD at desired locations.

Description

Please follow Diligent's PMOD IP tutorial to add this IP to the project (<https://reference.digilentinc.com/learn/programmable-logic/tutorials/pmod-ips/start>). Once added to the block diagram and exported to SDK. The header file needs to be imported to use the built APIs for the IP.

```
#include "PmodBT2.h"
```

The module also needs to be declared and instantiated before it is able to give readings.

```
PmodBT2 myDevice1, myDevice2;  
PmodMAXSONAR mySonar;  
SysUart myUart;
```

```
MAXSONAR_begin(  
    &mySonar,  
    MAX_SONAR_BASEADDRESS,  
    BT2_UART_AXI_CLOCK_FREQ);
```

The distance reading in inches can be retrieved as an unsigned 32-bit value through Diligent APIs.

```
u32 dist = MAXSONAR_getDistance(&mySonar);
```

Please see

(https://github.com/CaptainPenguins/SimpleAudioRecognition/blob/master/VivadoProject/Nexys-Video-Hub/sdk/hub_fpga/src/hub_control.c) for more details.