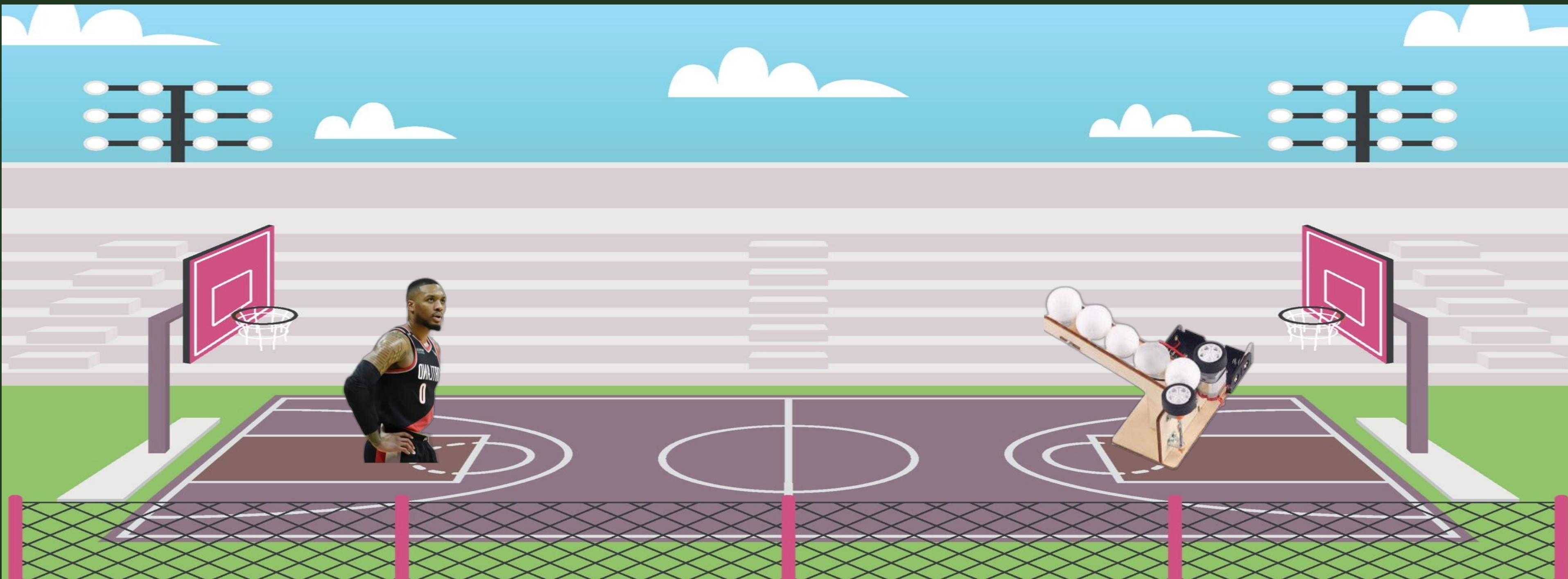


Interactive Basketball Launcher



Ibrahim Binmahfood, Robert Wilcox, Mohamed Gnedi

December 5, 2023



Agenda

Slide

- Overview
- Block Diagram
 - Joystick
 - Sensor
 - VGA Display
 - Servo & DC Motors
- Demo/Next steps

Presenter

- (Mohamed)
- (Mohamed)
- (Mohamed)
- (Ibrahim)
- (Ibrahim)
- (Robert)
- (Robert)

Overview

- **What is our Final Project?**

The subject project is an interactive game designed with RVfpga to provide users with three distinct features:

- i. User control
- ii. Mechanical output
- iii. Detection and display

- **How many players can play this game?**

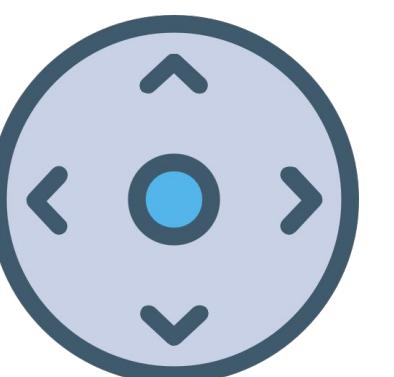
One player. However, To increase the number of players in a game:

- 1) Duplicate the control modules and connect them to the hardware
- 2) Manage scoring and player turns through software

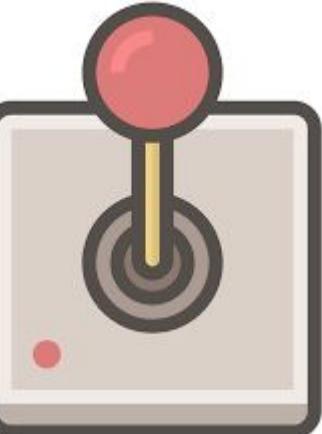
- **How to play this game?**

The game allows users to operate a ball launcher using a joystick and buttons to rotate and launch it. The ball is then detected using a sensor and the result is displayed.

User Control



Buttons



Joystick

Mechanical



Servo Motor



DC Motor

Detection



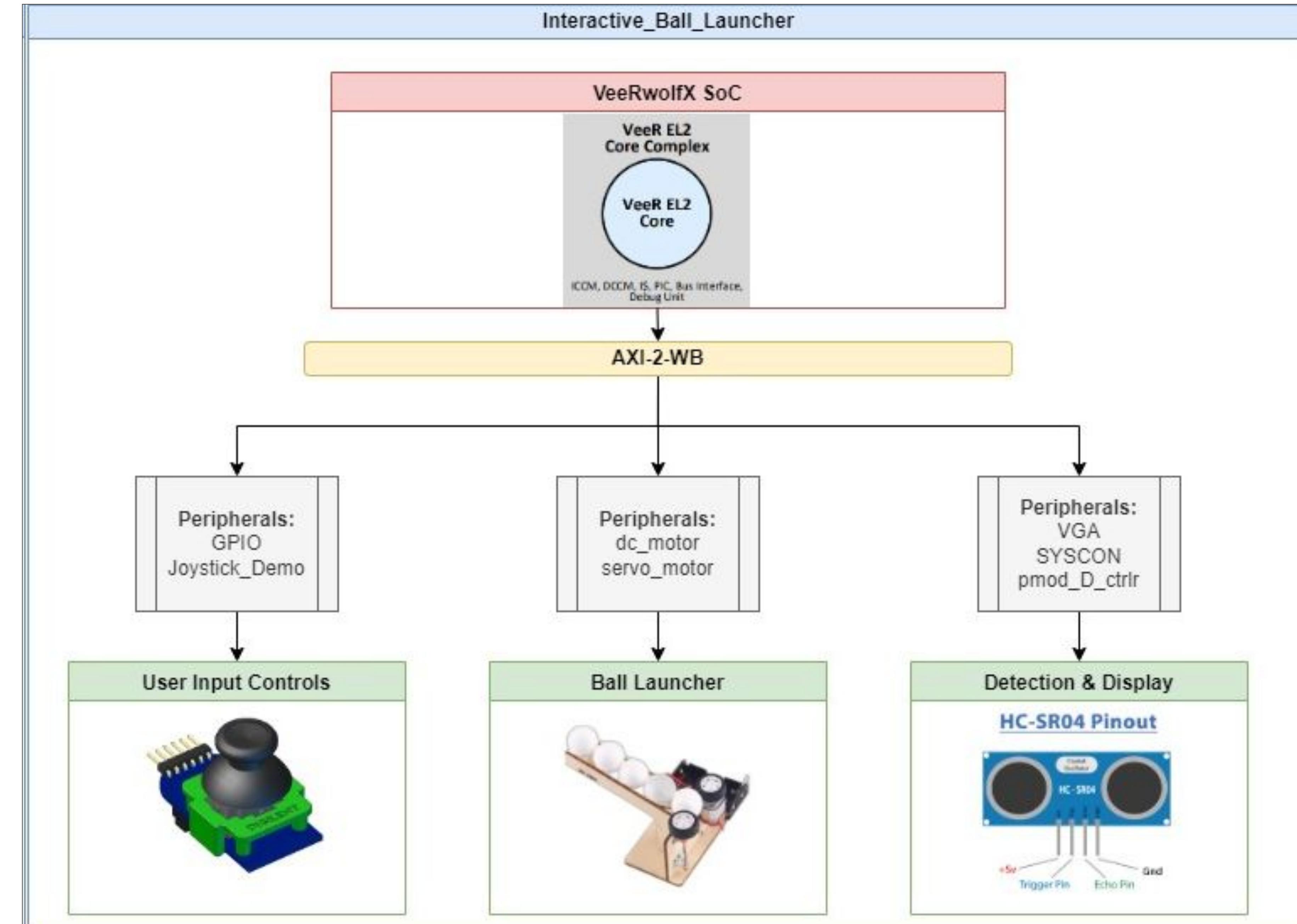
Sensor



VGA Display



Block Diagram



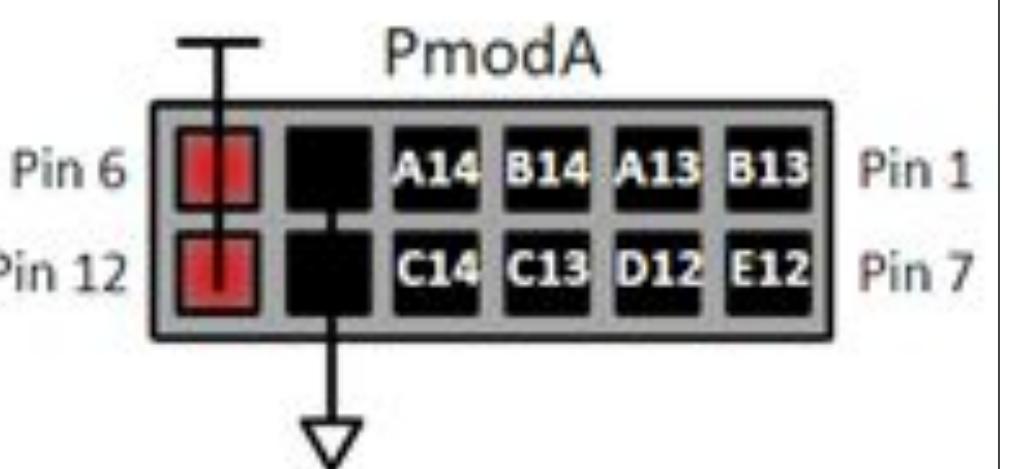
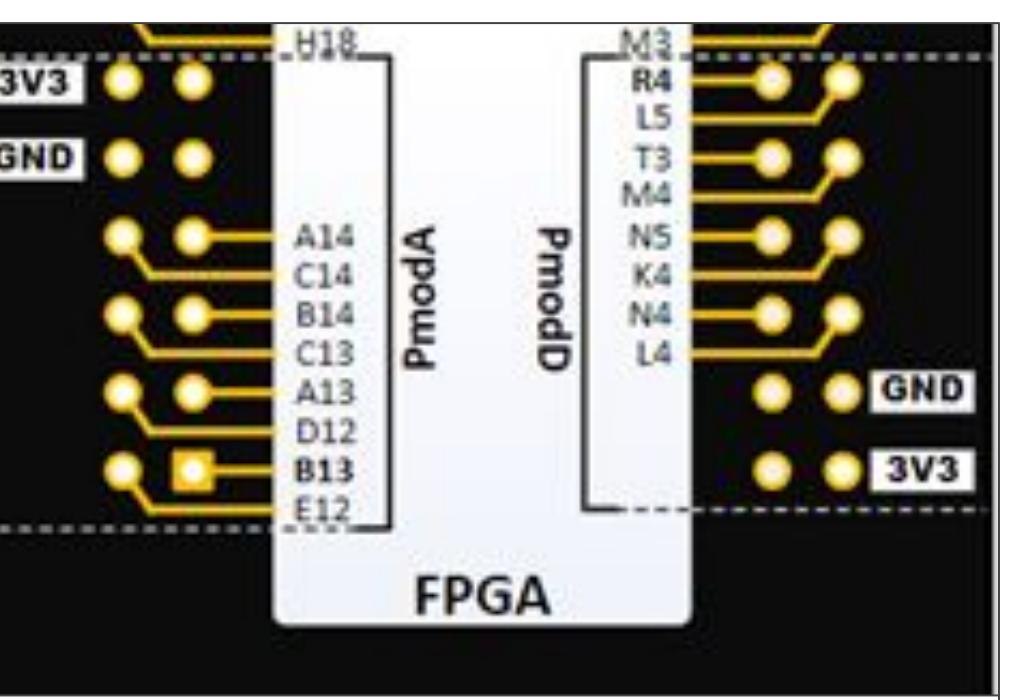
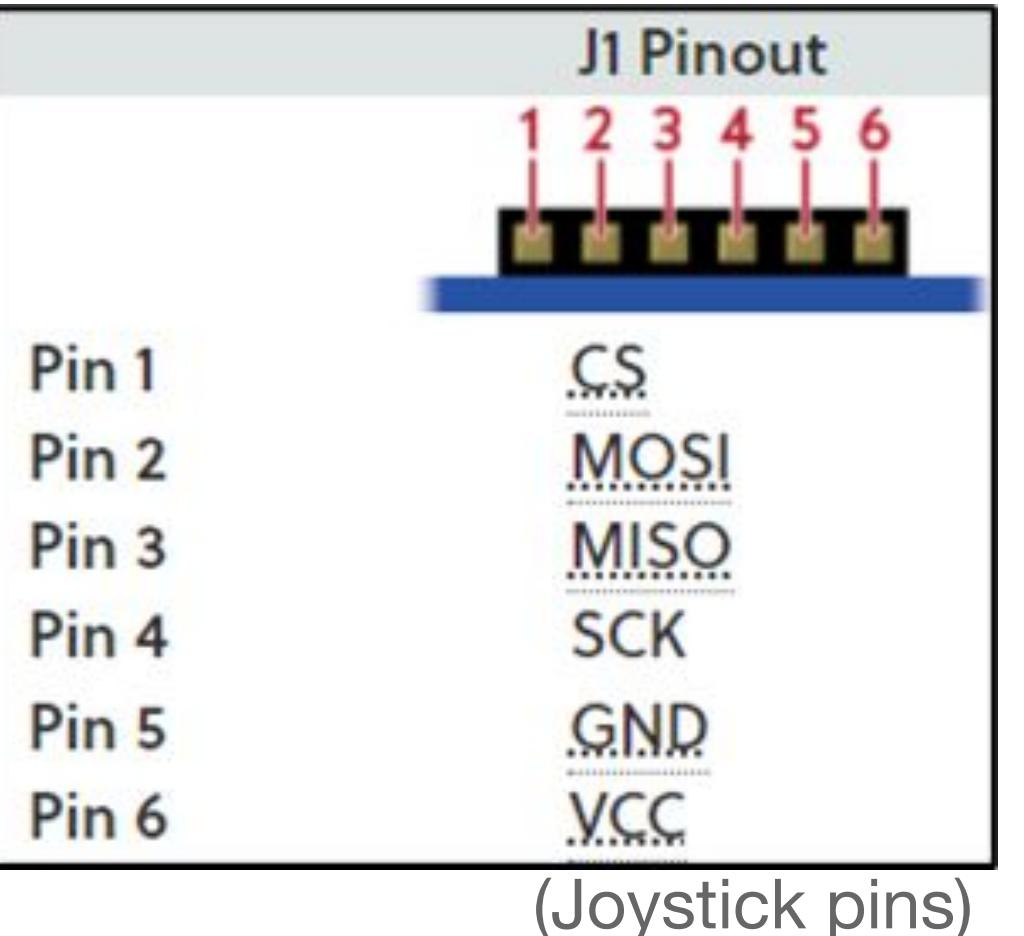
Joystick (1)

● Hardware Connections

- The **Pmod_Joystick** has 6 pins that are used for power, clocking, and communication.
- The pins are connected to the FPGA through PmodA lines.

● HDL Modules

- The **PmodJSTK_Demo** module communicates with the WB interface and utilizes 5 modules for SPI communication and transferring data.
- 5 sub-modules:
 - ClkDiv_5Hz.v (100MHz -> 5Hz Clock divider)
 - PmodJSTK.v
 - ClkDiv_66_67Hz.v (100MHz -> 66.67kHz Clock divider)
 - spiCtrl.v (SPI FSM)
 - SPImode0.v (SPI Mode 0 controller)

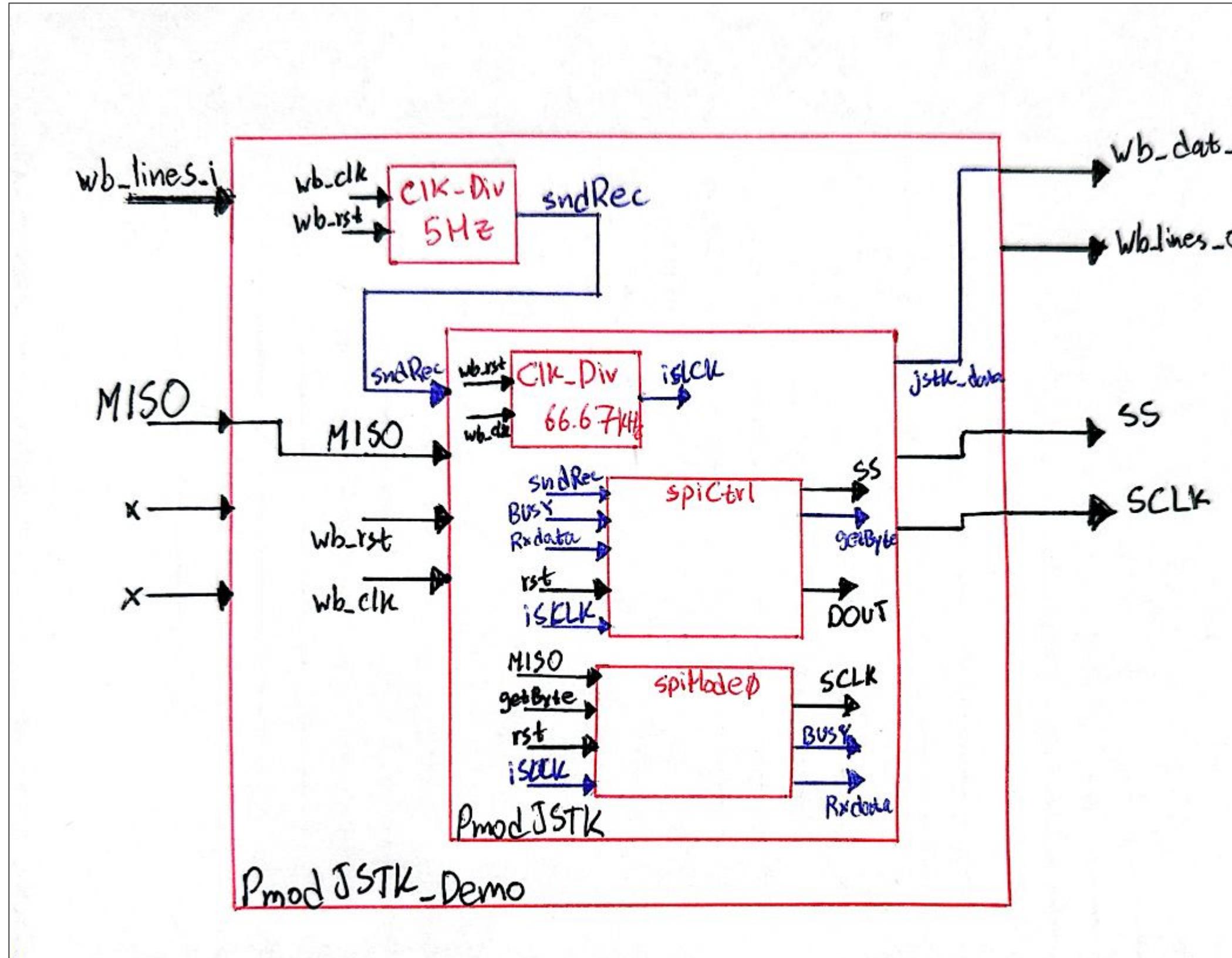


(FPGA Pmod pins)

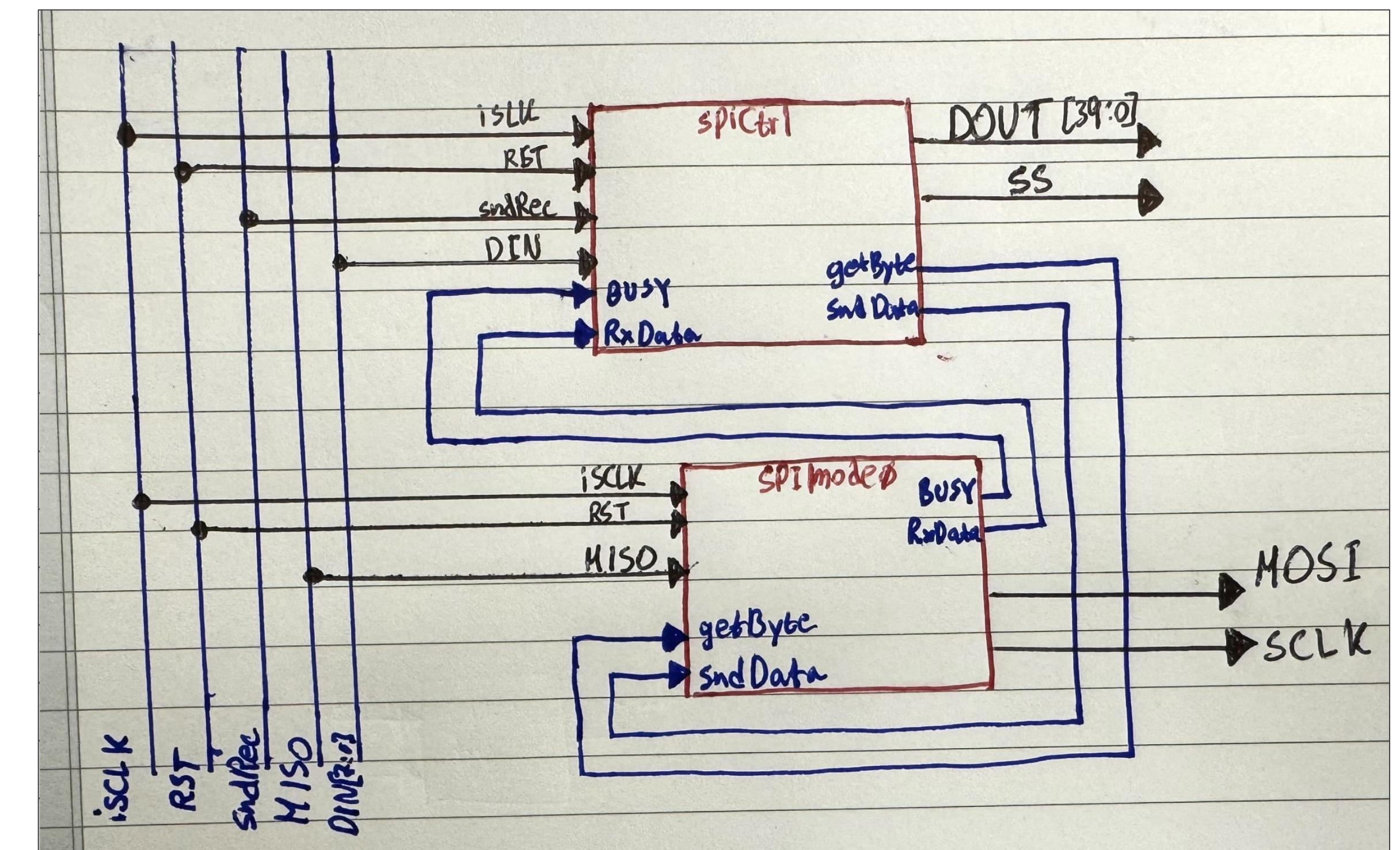


Joystick (2)

- Pmod_JSTk_Demo.v



- spiMode0 & spiCtrl

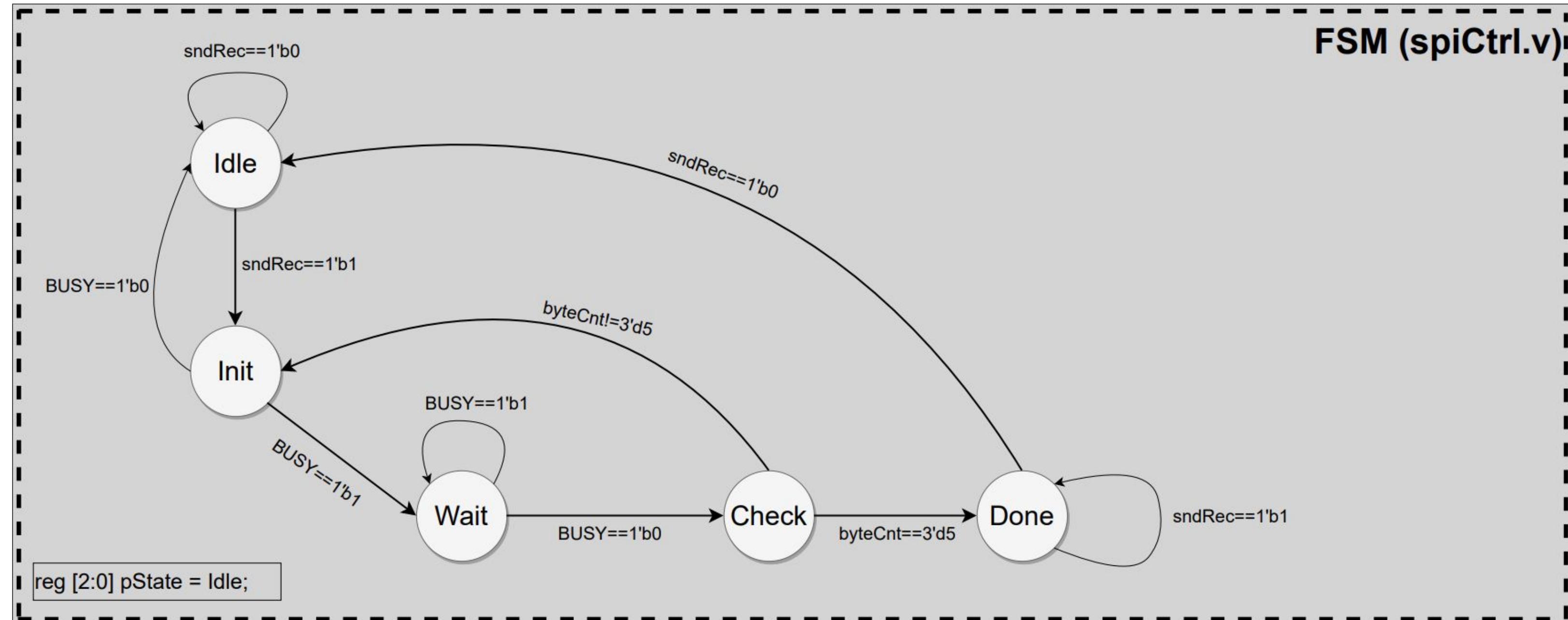




Joystick (3)

- **spiCtrl FSM**

User sends a zero and a series of 4 dummy bytes to receive the standard 5 bytes of data



- **Interfacing with PMOD**

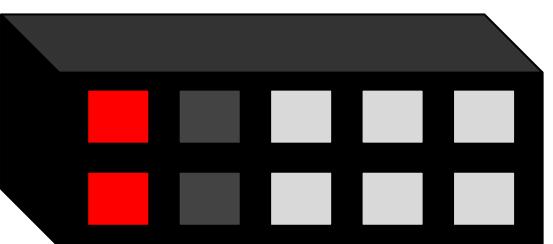
- The two bytes of data for each axis represent the 10-bit value for that axis.
- For the x-axis, a 0 value corresponds to the axis being tilted fully to the left and a value of 1023 corresponds fully to the right.

	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
MOSI	COMMAND / 0	PARAM1 / DUMMY	PARAM2 / DUMMY	PARAM3 / DUMMY	PARAM4 / DUMMY
MISO	smpX (Low Byte)	smpX (High Byte)	smpY (Low Byte)	smpY (High Byte)	fsButtons

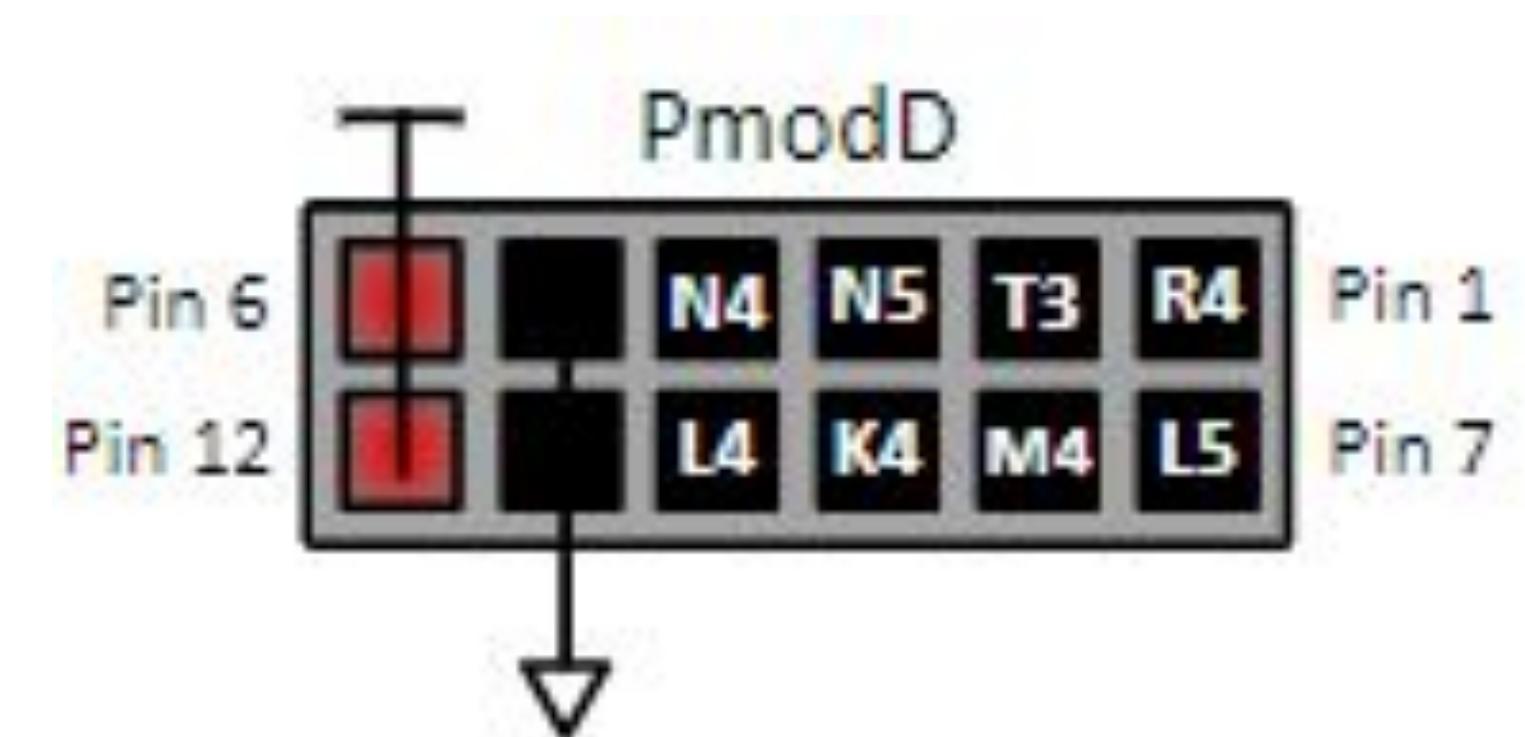


Proximity Sensor (1)

Wishbone Peripheral Module - *pmod_D_ctrlr.sv*



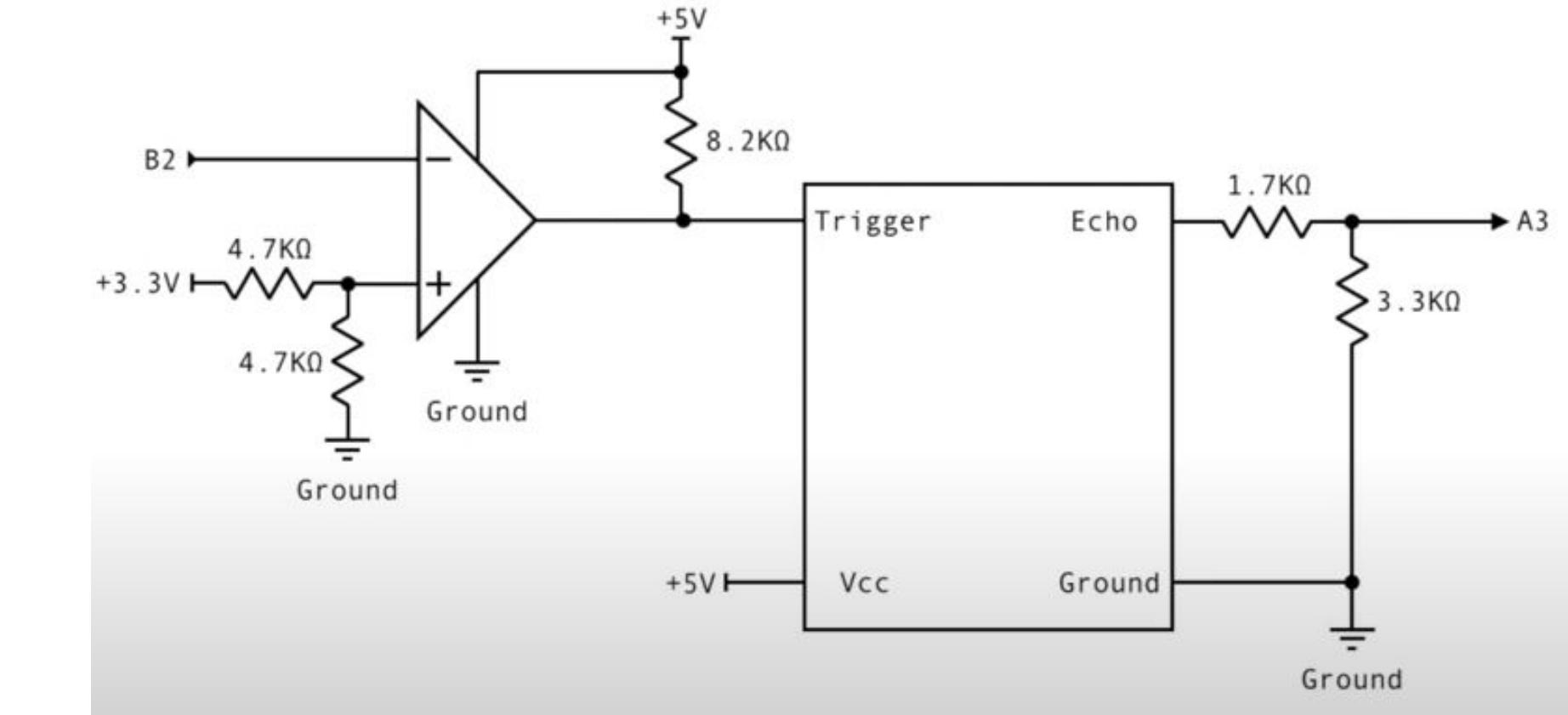
- PMOD D Connector (Left to Right): 3.3V, GND, I/O signals
- I/O signals connections:
`rvfpガboolean→veerwolf_core→pmod_D_ctrlr→ultrasonic_sensor`



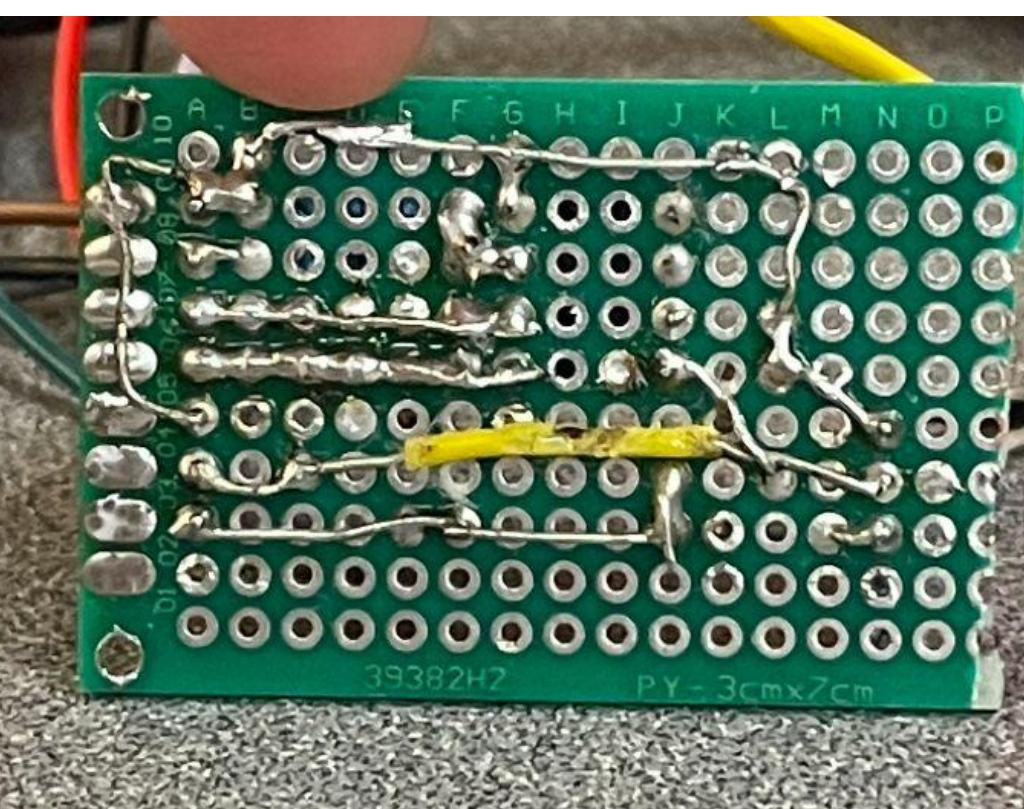
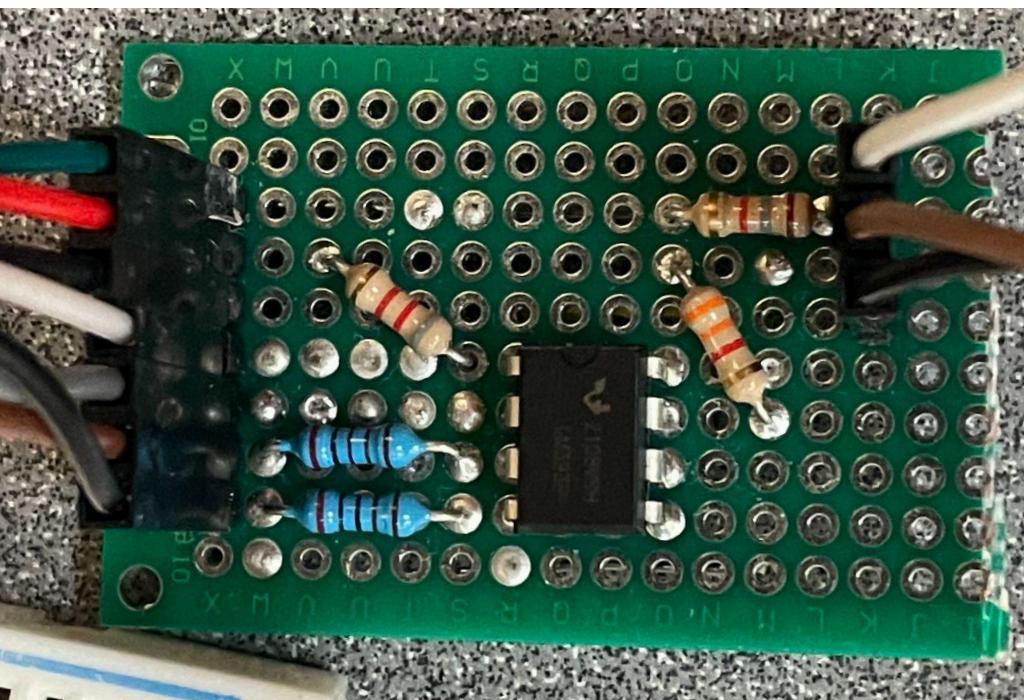
- PMODD_STATE Register: READ only
 - bit[31:2] - reserved
 - bit[1:0] - state at bit 1 and echo_pulse at bit 0
- The state bit is an output from an OR filter. The echo_pulse bit comes from the input to the PMOD D expansion port.
- This register is located at 0x80001640.



Proximity Sensor (2)



Interface Circuitry for the HC-SR04



CLK IP for *ultrasonic.sv*

- Requires 64 MHz clock to bit select a counter register that is incremented every clock edge. This will take a 10 us pulse output to the sensor *Trigger* pin.

ultrasonic.sv RTL Module

- Distance Limit set ~10cm
- Trigger Pulses are bit slices of the counter
- OR filter filters false positive echo pulses
- Sample Rate is a bit slice from a counter



VGA Display

VGA Firmware *vga.h* / *vga.c*

- Utilizes a delay between writing characters so that by the naked eye multiple characters are visible
- Characters are written one by one with set positions for a welcome, hit/miss, score and screens
- Available ASCII characters to display:
 - [a-z]
 - [0-9]
 - [!, #, “, space]

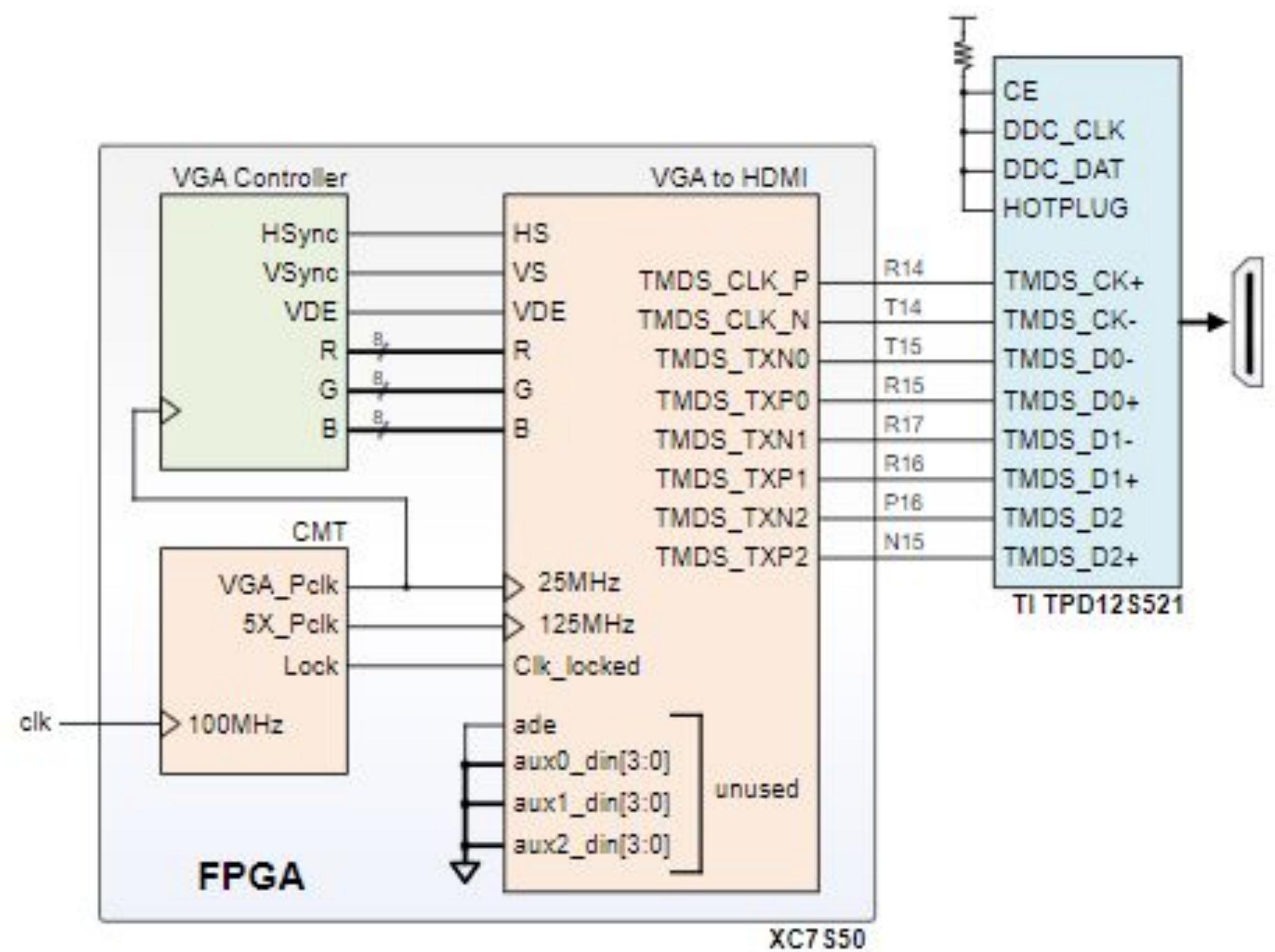
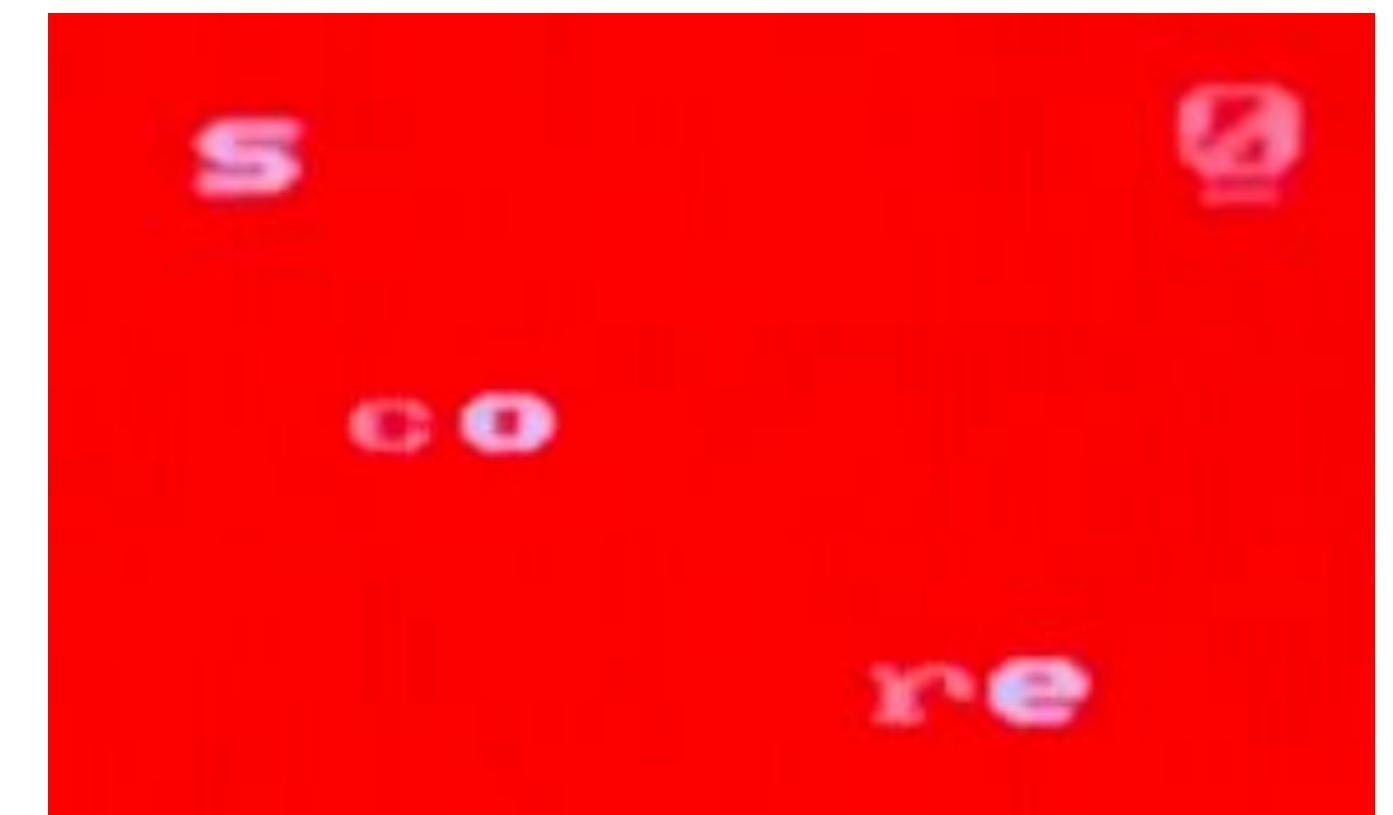


Figure 1. Block diagram of VGA to HDMI system





Ball Launcher

Ball Launcher hardware

- The ball Launcher used was purchased from Xplorably
- The firing mechanism for the ball was powered with a 5V DC hobby motor
- An SG90 Stepper motor was used for the aiming control





Ball Launcher

Servo controller peripheral module

Design Elements

- PWM Logic: Controls servo motor via PWM signal
- Wishbone Interface: Supports configuration via Wishbone signals
- Utilizes the BooleanBoard's onboard servo pins



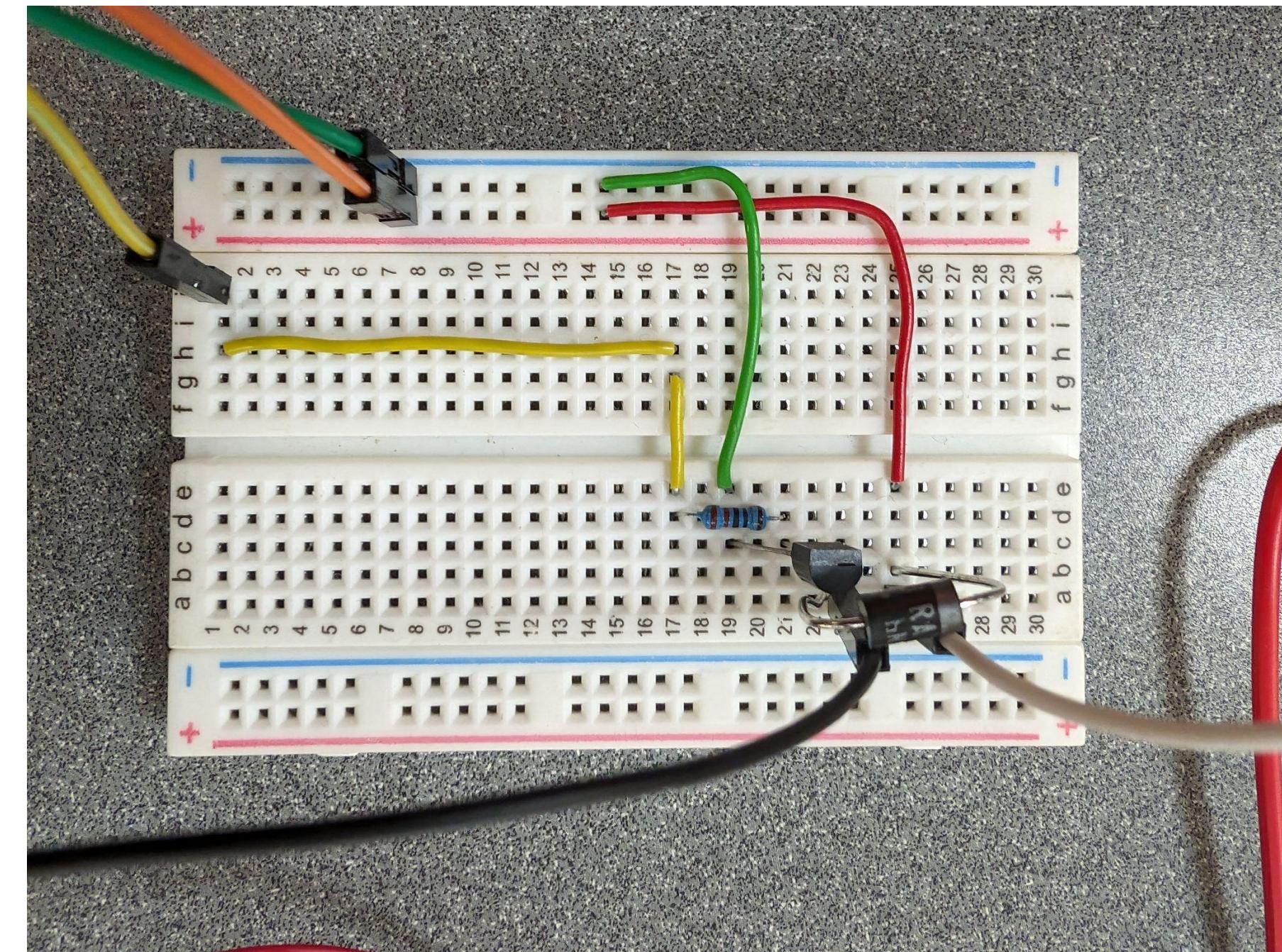


Ball Launcher

DC Motor controller peripheral module

Design Elements

- PWM Logic: Controls DC motor via PWM signal
- Wishbone Interface: Supports configuration via Wishbone signals
- Utilizes the BooleanBoard's onboard servo pins



- MOSFET switch circuit designed to power the DC motor with board power
- Simple NPN transistor with a schottky diode to protect against inductive kick



What we learned & Next Steps!

- Developing these modules was a great way to further our knowledge of FPGAs, specifically the VeerWolf core
- Developing separate modules which we then had to integrate into one hierarchy presented yet unseen challenges, but it was a great learning experience (especially because we were working on different platforms)
- We learned that it may be wise to set less loft goals, this was a real challenge to get done in the timeframe we had



Demo Time
Are you ready?



Thank You! Questions?

