Chapter 6:
Game Controller
Hardware

Originally I was going to use Atari 2600 joysticks for the HYDRA; however, due to their scarcity and the lack of extra controls on them, the decision was made to go with classic NES (Nintendo Entertainment System) controllers. The primary reasons are that NES controllers are abundant, easy to interface to, have lots of nostalgia, can be interfaced via 4 signal lines, third-party models look really cool, and most importantly, if the HYDRA interfaces to the NES controller then any "NES compatible" device can be interfaced since they all use the same connector! Thus, light guns, VR gloves, and any and all manner of bizarre NES hardware from the 80's/90's can be interfaced to the HYDRA through the controller ports, so it's a good plan. With that in mind, in this chapter we are going to discuss the following topics:

- ▶ Overview of the NES hardware
- ▶ NES controller protocol and interfacing
- ▶ Supporting Super NES controllers

6.1 The Classic NES Controller (1983)

If you have never heard of Nintendo, then may I be the first to greet you from Earth, because you must be from another planet! The Nintendo Entertainment System or NES, also known as the Nintendo Family Computer (Famicom) was released in Japan in 1983 and shortly after in the USA in 1985. The system was even more successful than the Atari 2600 and more or less put the last stake in the heart of Atari home consoles forever. The NES sold more than **60,000,000** units worldwide in its heyday and still sells to this day (I just bought 4 of them from eStarland.com)! There are over 200 clones of the system in circulation and the word "Nintendo" is used in many cultures as a youth slang term with many "varied" meanings.

The system itself was nothing compared to the Propeller chip's computing power, the NES was powered by the 8-bit 6502 running at 1.79 MHz, but the NES did have quite a bit of dedicated graphics hardware making it quite a contender in the 8-bit era, it sported multiple tile maps, graphics planes, sprites, sound, and a lot more. In any event, as noted I opted to use the NES controllers for the aforementioned reasons, plus being an Atari guy, sometimes its nice to jump ship and see how the other side lives, so all in all I am very happy with the NES controllers on the HYDRA. They give the system a retro feel, but are powerful enough to use as serious input devices for more than just games. Plus, the final controllers shipped with the HYDRA look very cool.
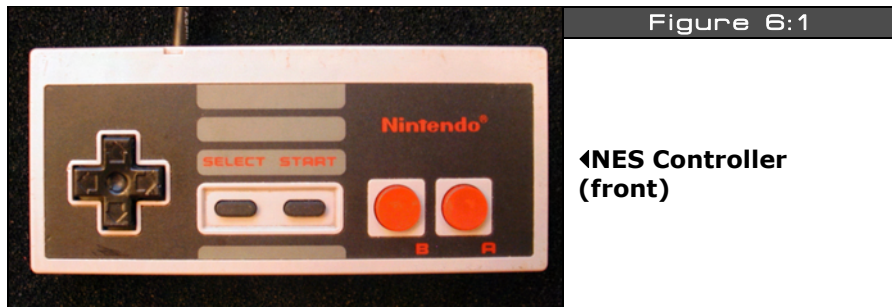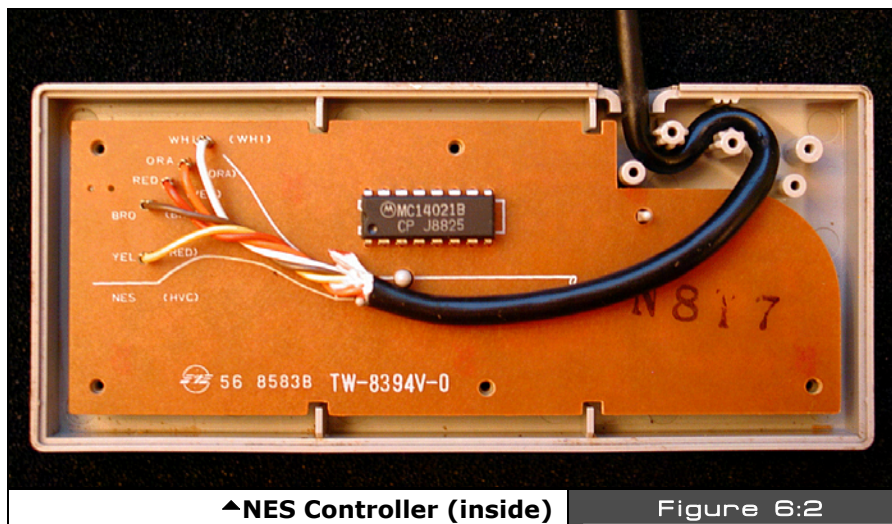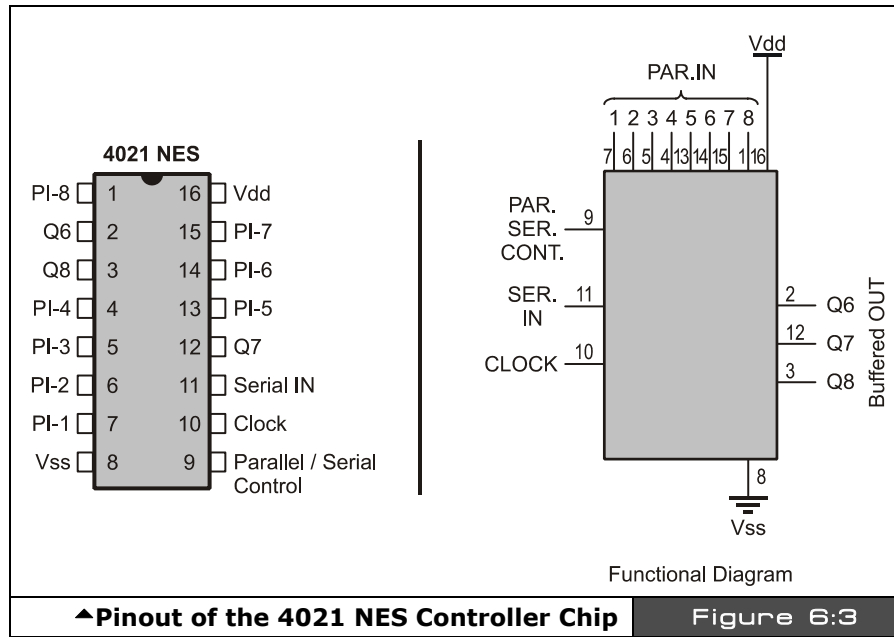
◂NES Controller
(front)

The classic NES controller / gamepad is shown in Figure 6:1, and internal shot is shown in Figure 6:2. As you can see there isn't much to the NES controller electronics (a single 4021 serial chip), but that's the beauty of it. The interface is so simple, we can use 4 lines to interface to the NES controller and read all 8 inputs (Up, Down, Left, Right, A, B, Select, Start).

Of course, we aren't using these collectors' item controllers on the HYDRA, that would be sacrilege; but all NES controllers must be compatible with the original standard, so we are going to discuss NES controllers with the original in mind, but the controller that comes with your HYDRA will be from the 21st century and a bit more sleek in its design.
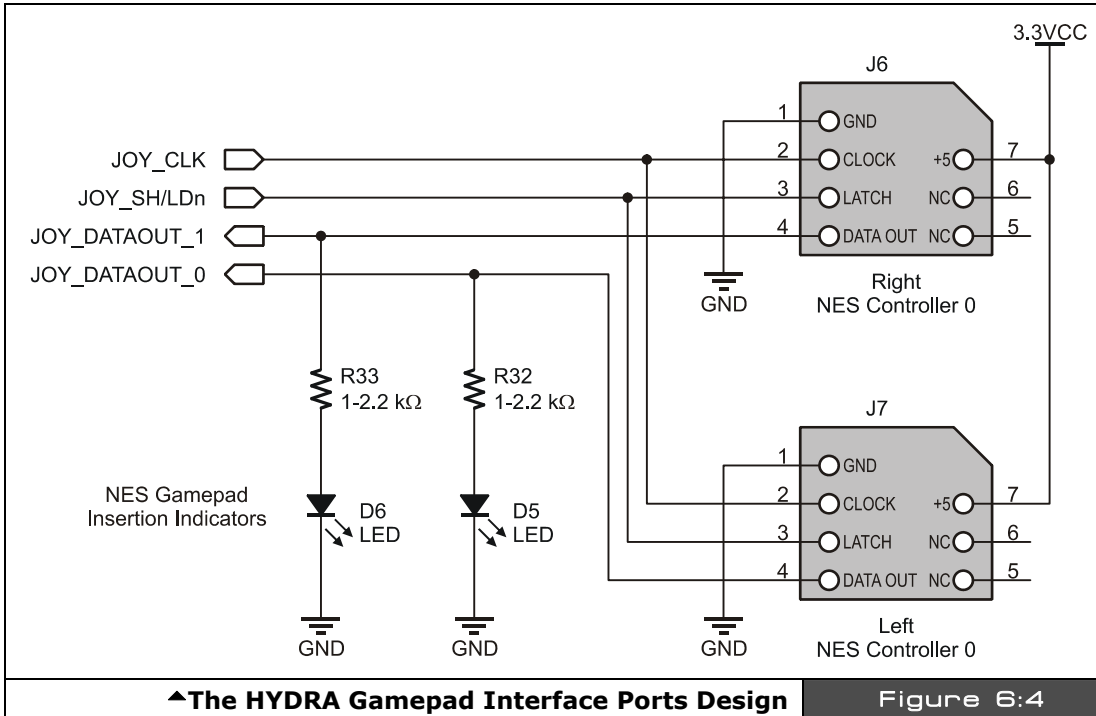

▴NES Controller (inside)   Figure 6:2

4021 NES

PI-8	1	16	Vdd
Q6	2	15	PI-7
Q8	3	14	PI-6
PI-4	4	13	PI-5
PI-3	5	12	Q7
PI-2	6	11	Serial IN
PI-1	7	10	Clock
Vss	8	9	Parallel / Serial Control

Functional Diagram

▲Pinout of the 4021 NES Controller Chip	Figure 6:3

The NES controller is based on a parallel to serial shift register internally that "latches" the button states and then shifts them out to the host. The shift register used in the original NES control is the *CMOS 4021 shift register*. Figure 6:3 shows the pinout of the 4021 and the complete data sheet is on the CD located at:

CD_ROOT\HYDRA\DOCS\cd4021b.pdf

Game Programming for the Propeller Powered HYDRA ‣ **Page 97**

Figure 6:4 shows the HYDRA controller interface for the right and left controllers. As mentioned, all the signals are paralleled except for the JOY_DATAOUT_0 and JOY_DATAOUT_1 lines since these are the independent data streams from each controller. Also, notice that the LEDs that are connected to the data out lines, the resistors are large enough to minimize current drain, and thus the signals are not degraded, but when the NES controllers are plugged in the weak pull ups on the outputs of these lines power the LEDs and indicate that the controller is plugged in.

▲The HYDRA Gamepad Interface Ports Design   Figure 6:4

The actual pinouts and colors (for the classic NES controller) are shown in Table 6:1. Referring to the table, you can see that the original spec for the NES controllers uses +5 for power. I was a bit worried that the old 4021's wouldn't work at 3.3 V and I would have to Frankenstein the 5 V supply into the NES controller, but then make sure the levels match the outputs and inputs to interface to the Propeller, but the 4021 is a CMOS device and works fine at 3.3 V. Of course, running any CMOS device rated for 3-7 V at lower voltages lowers the maximum clocking rate, but in this context it's irrelevant, since we can read the controller bits at a rate of 200 ns each.

Table 6:1			NES Controller Pinout▾
Pin #	**Function**	**Wire Color***	**Pin on Propeller Chip/Design Designator**
1	Ground	Brown	GND
2	Clock	Red	JOY_CLK (P3)
3	Latch	Orange	JOY_SH/LDn (P4)
4	Data Out	Yellow	JOY_DATAOUT_0 (P5), JOY_DATAOUT_1 (P6)**
5	No Connect	None	
6	No Connect	None	
7	+ 5 Power	White	
Notes	* Colors on original NES Nintendo controllers only; 3rd party controllers have different colors. ** Where JOY_DATA_OUT_0|1 are the left and right controller ports respectively.		

NOTE

*These colors are on "stock" controllers ONLY, in fact many third-party manufacturers of NES/SNES controllers not only get the colors wrong, but sometimes use the correct colors and connect them to the wrong pins! The best thing to do is rip the connector apart and see what color is connected to what pin on the 4021 and power supply rails.

6.2   Interfacing to the NES Controllers

Interfacing to the NES controllers is a completely digital affair; on the HYDRA for example, we can run the clock, latch, and power lines in parallel, then send the data from each controller (if its plugged in) to the Propeller chip. To read the state of the NES controllers, you must perform the following steps:

Step 1: Set the Latch line (P3 on Propeller chip) to LOW (this is connected to the parallel/serial select on the 4021).

Step 2: Set the Clock to LOW.

Now, we are in a known state.

Step 3: Pulse the Latch line HIGH for at least 200 ns, this will latch the data into the shift register.

Step 4: Read the bits in from the Data Out lines at P5 and P6, the first bit will be available after the latching process. To read the remaining bits, pulse the Clock line 7 times, as you are doing this you can shift each bit read from the Data Out line into a BYTE, so that the first bit ends up in the bit7 position and the last bit in the bit0 position. Realize that the HYDRA reads the controllers in parallel, so you can perform the read operation once with some clever bit manipulation.

Step 5: The button states are all inverted; that is, pressed is digital 0, released is 1, so you might want to invert the results so 1 means "pressed", just personal preference.

After performing these steps you simply use bit operations to determine which buttons are pressed. The button encodings are shifted out in the following sequence starting from left to right, where the left-most bit is available on the data out lines after the initial latching operation as shown in Figure 6:5.
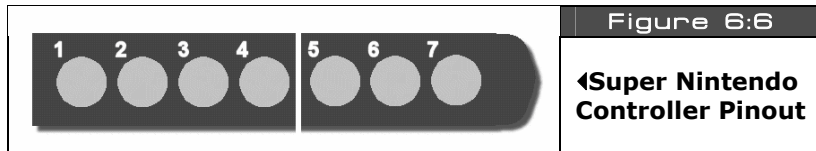
Operation	Latch	CLK	CLK	CLK	CLK	CLK	CLK	CLK
	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7
Data	A	B	Select	Start	Up	Down	Left	Right

Figure 6:5

◀NES Data Format

Also, a bonus feature of the NES controller is that when you read it on the HYDRA, if the controller is NOT plugged in then the value $FF will be returned, thus we can use this sentinel to determine if a controller is plugged in or not!

> **WARNING** When you latch and clock the bits, remember that the 4021 chip has a maximum clocking rate. I suggest keeping the clocks 200 ns or greater to allow for settling and propagation delays, you don't want to loose the fire button bit under a high stress condition if you are reading the controller too fast!

6.3   Super NES Controller Interfacing (1990)

Again for a quick history lesson: the new 16-bit system, Super Nintendo (SNES), was released in Japan in 1990 and 1991 in the USA. The system once again became an instant hit and sold a total of 50,000,000 units worldwide, thus making the combined sales for NES and SNES in excess of 100,000,000 units!!! The SNES had better graphics and a much more powerful 65816 processor (basically a 16-bit version of the 6502) running at 1.79 – 3.58 Mhz in a "variable" speed mode of operation. The overall design of the SNES was a little more sleek and updated, and of course they updated the controllers as well. The new controller has a few more buttons, but the underlying protocol for reading the controller is the same. Additionally, the mechanical interface for the SNES controller (shown in Figure 6:6) is even more bizarre than the NES controller.

The Super NES controller is almost identical to the NES controller as far as the protocol goes, so the techniques here are applicable to the SNES controller as well. Figure 6:6 shows the pinout of the Super NES controller and Table 6:2 lists the pinout.

Table 6:2	Super NES Controller Port Pinouts▾	
Pin #	**Function**	**Wire Color**
1	+ 5 power	White
2	Clock	Yellow
3	Latch	Orange
4	Data Out	Red
5	No Connect	None
6	No Connect	None
7	Ground	Brown

The only difference is that the Super NES streams out 4 more bits representing X, Y, R (right shoulder button), and L (left shoulder button) which makes for a total of 12 bits. However, Nintendo opted to add 4 more cycles for future expansion and made the read protocol a total of 16 bits as shown in Figure 6:7; notice the last 4 bits are always HIGH, or 1s.

Operation	Latch	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK
	bit 0	bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7	bit 8	bit 9	bit 10	bit 11	bit 12	bit 13	bit 14	bit 15
Data	B	Y	Select	Start	Up	Down	Left	Right	A	X	L	R	1	1	1	1

▲**SNES Data Format** Figure 6:7

6.4   Summary

In conclusion, the NES controllers are pretty slick internally since they simplify the HYDRA's need for support hardware due to their built in serializing chips. Also, hopefully you can appreciate how simple and brilliant they were. A piece of software or hardware is seen through its interfaces or GUI; this is true with Windows and true with game consoles, so considering that, any idea that helped sell 100,000,000 units is worth taking a closer look at!