Rangefinder IP Manual for Touch-Free Space Invaders

Luke Hsiao & Jeff Ravert

Brigham Young University

21 November 2014

Ultrasonic Rangefinder and Photoresistor Descriptions

This lab used the HC-SR04 ultrasonic rangefinder to control the movements of the tank in Space Invaders and a photoresistor to control the firing of the tank. In order to interface the HC-SR04 and photoresistor with the current hardware system, a new IP was created and added to the system.

The HC-SR04

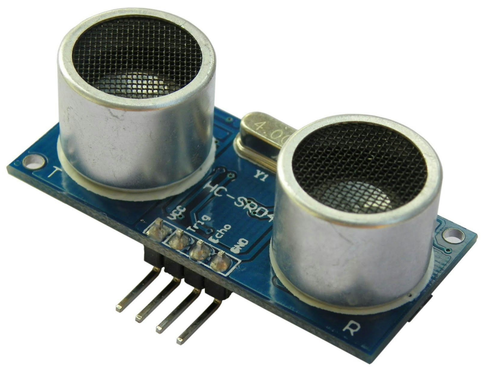

Figure 1: Cytron HC-SR04

The HC-SR04 is manufactured by Cytron Technologies. It uses sonar to determine distance to an object from 2cm to 400cm (1 inch to 13 feet). Its operation is not affected by sunlight or black material. However, it can struggle reading acoustically soft materials like cloth. The HC-SR04 has four pins (1) VCC, (2) Trigger, (3) Echo, and (4) GND. It is run of 5 V for Vcc.

Operation is described by the timing diagram below:

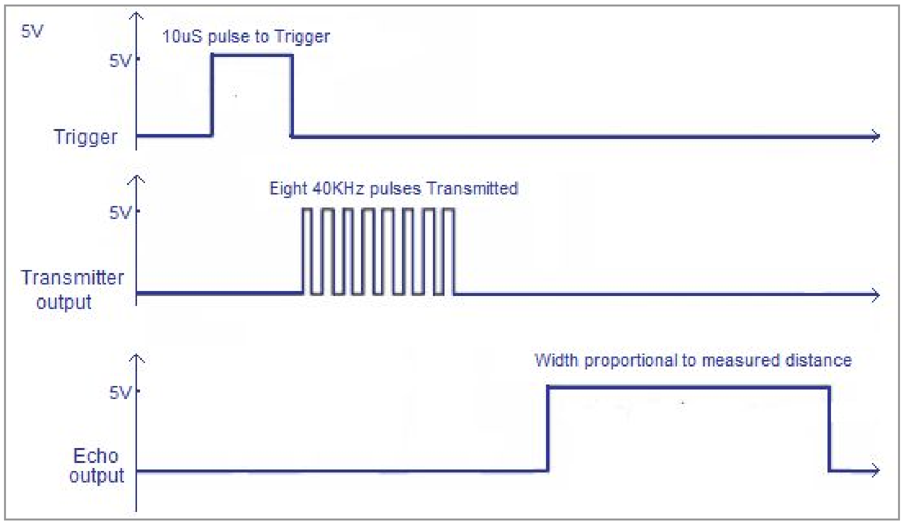
5V
10uS pulse to Trigger
5V
Trigger
Eight 40KHz pulses Transmitted
5V
Transmitter output
Width proportional to measured distance
5V
Echo output

Figure 2: HC-SR04 Timing Diagram

To start measurement, the Trigger of the HC-SR04 must receive a pulse of high (5V) for at least 10μs, this will initiate the sensor will transmit out eight cycles of ultrasonic burst at 40kHz and wait for the reflected ultrasonic burst. When the sensor detects ultrasonic sound reflecting back, it will set the Echo pin to high (5V) and delay for a period (width) that is proportional to distance. To obtain the distance, measure the width of pulse on the Echo pin. Distance can then be calculated as follows:

Time = Width of Echo pulse, in μS

- Distance in centimeters = Time / 58
- Distance in inches = Time / 148
- Or you can utilize the speed of sound, which is 340m/s

The Photoresistive Sensor

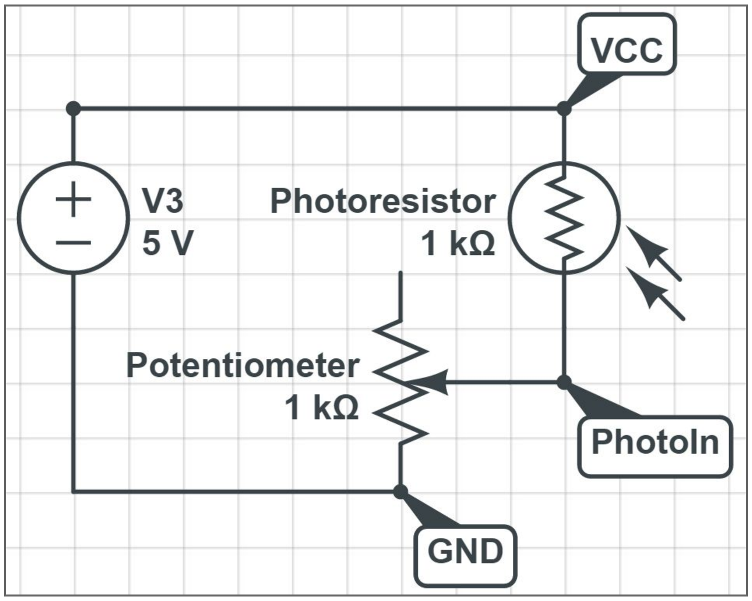

Figure 3: Photoresistor Schematic

In order to allow a player to fire the tank bullet my waving his or her hand, a photoresistor circuit was used. In conjunction with a simple voltage divider, the photoresistor sends a logical low or high signal to the Rangefinder IP core.

Using the same 5V VCC as the HC-SR04 and same GND, the voltage divider allows PhotoIn in the schematic to be tied to the photoIn pin of the Rangefinder IP core (Figure 4). The voltage levels can be fine-tuned using the potentiometer so that a logical 1 occurs when light is hitting the diode, and a logical 0 occurs when light is being blocked.

The Rangefinder IP Core

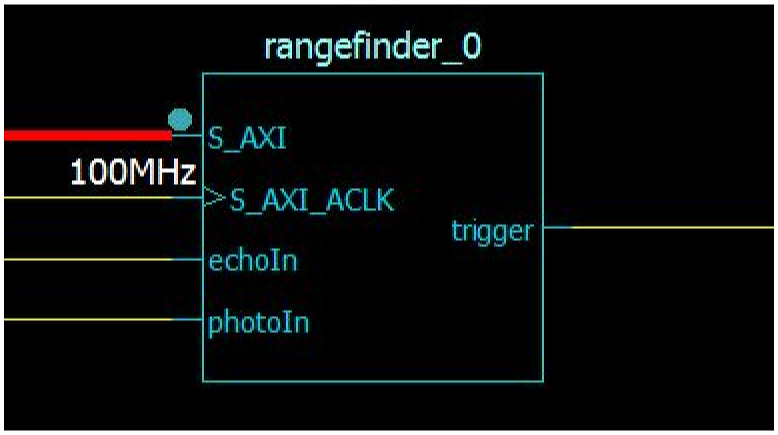

Figure 4: Rangefinder IP Block Diagram

In order to interact with the HC-SR04 and Photoresistor, a new IP module was created: the Rangefinder IP. The Rangefinder IP is a simple state machine. At regular intervals, it sends a 10µs trigger pulse, then listens and counts the width of the echo signal in clock cycles. This width value is then stored in a software-accessible, read-only register. It also synchronizes the asynchronous photoIn input from the Photoresistor circuit, and stores the value in a second read-only register.

The software is able to access these registers by addressing an offset of the Rangefinder's base address. In this implementation of the hardware, the base address and high address of the Rangefinder are defined as:

```
#define XPAR_RANGEFINDER_0_BASEADDR 0x76400000
#define XPAR_RANGEFINDER_0_HIGHADDR 0x7640FFFF
```

Table 1 shows the offset that must be added to the base address of the PIT to access the register. For convenience, and API has been created and described below that provides a simple macro for writing to these registers.

Register Name	Offset from Base Address	Function
Distance Register	0x0	Always contains the most recent distance read by the HC-SR04. This register is READ ONLY.
Photo Register	0x4	Contains 0xFFFFFFFF if light is getting to the photo resistor. If light is blocked, it contains 0x0. This register is READ ONLY.
Register 2	0x8	Reserved
Register 3	0xC	Reserved

Table 1: General Register Descriptions

Distance Register

Inside the Rangefinder IP, a counter is set to zero and then begins incrementing while the HC-SR04's ECHO signal is high. The system uses a 100MHz clock, so each clock cycle corresponds to 10ns. The counter increments until ECHO drop low, then stores the value into the distance register. This value is generally between 18,500,000 and 12,000, which correspond to the maximum and minimum time that the ECHO output is set high by the HC-SR04. Consequently, this 32-bit counter will never overflow.

Photo Register

While the Distance Register can be used to control tank movement, the Photo Register can be used to fire bullets in Space Invaders. This register simply contains all 1's if the photo resistor is receiving light, or all 0's if light is blocked (signaling that the bullet should be fired). Note that the Photo Register will constantly contains 0's as long as light is being blocked and constantly contain 1's while light is visible.

Timing Diagrams

The timing diagram below shows the typical operation of the Rangefinder.

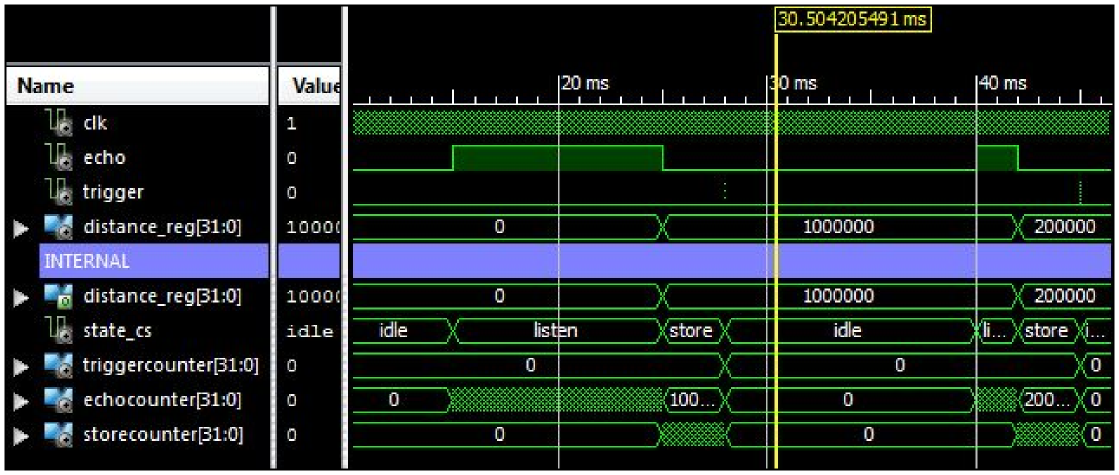
30.504205491 ms
Name
Value
20 ms
30 ms
40 ms
clk 1
echo 0
trigger 0
distance_reg[31:0] 1000(0 1000000 200000
INTERNAL
distance_reg[31:0] 1000(0 1000000 200000
state_cs idle idle listen store idle li... store i...
triggercounter[31:0] 0 0 0 0
echocounter[31:0] 0 0 100... 0 200... 0
storecounter[31:0] 0 0 0 0

Figure 5: Rangefinder Timing Diagram

As described above, the Rangefinder IP sends a short TRIGGER pulse, and waits for the ultrasonic sensors response. Then, it listens to how long the ECHO signal is, and stores the number of clock cycles into the Distance Register.

Note that because the HC-SR04's reading contain a small margin of error, the distance values store in the Distance Register will not be perfectly consistent. Proper software sampling techniques should be used to average the samples for smoother data.

Driver API

The Application Programming Interface defined to interact with the Rangefinder is outlined below. Although Xilinx's XPS generated additional macros, these are the only ones important to the operation of the Rangefinder. These macro definitions can be found in rangefinder.h.

RANGEFINDER_readDistance(BaseAddress)

This macro returns the 32-bit unsigned value currently stored in the Distance Register. The BaseAddress parameter is simply the base address assigned by Xilinx XPS of the Rangefinder.

RANGEFINDER_readPhoto(BaseAddress)

This macro returns the 32-bit unsigned value currently stored in the Photo Register. The BaseAddress parameter is the base address assigned by Xilinx XPS of the Rangefinder.

An example Initialization and use of the API in application code is shown below:

```
#define XPAR_RANGEFINDER_0_BASEADDR 0x76400000
...
unsigned int latestDistance
unsigned int fire;
...

// Read the Distance and photo
latestDistance = RANGEFINDER_readDistance(XPAR_RANGEFINDER_0_BASEADDR);
fire = RANGEFINDER_readPhoto(XPAR_RANGEFINDER_0_BASEADDR);
// manipulate tank based on latest distance
if (fire == 0) {
       //fire tank bullet
}
```

Once read, operations can be performed using latestDistance. Note that the Rangefinder IP is constantly updating the distance register and the photo register. Because of it's simplicity, no control registers are used to modify its operation. The application developer should keep this in mind so that proper smoothing algorithms can be used with the sampled data. For example, this implementation uses a moving average to smooth the samples read from the distance register to minimize erratic behavior.

System Setup

The Rangefinder IP was just added to our overall system by connecting it to the AXI Bus as shown below:

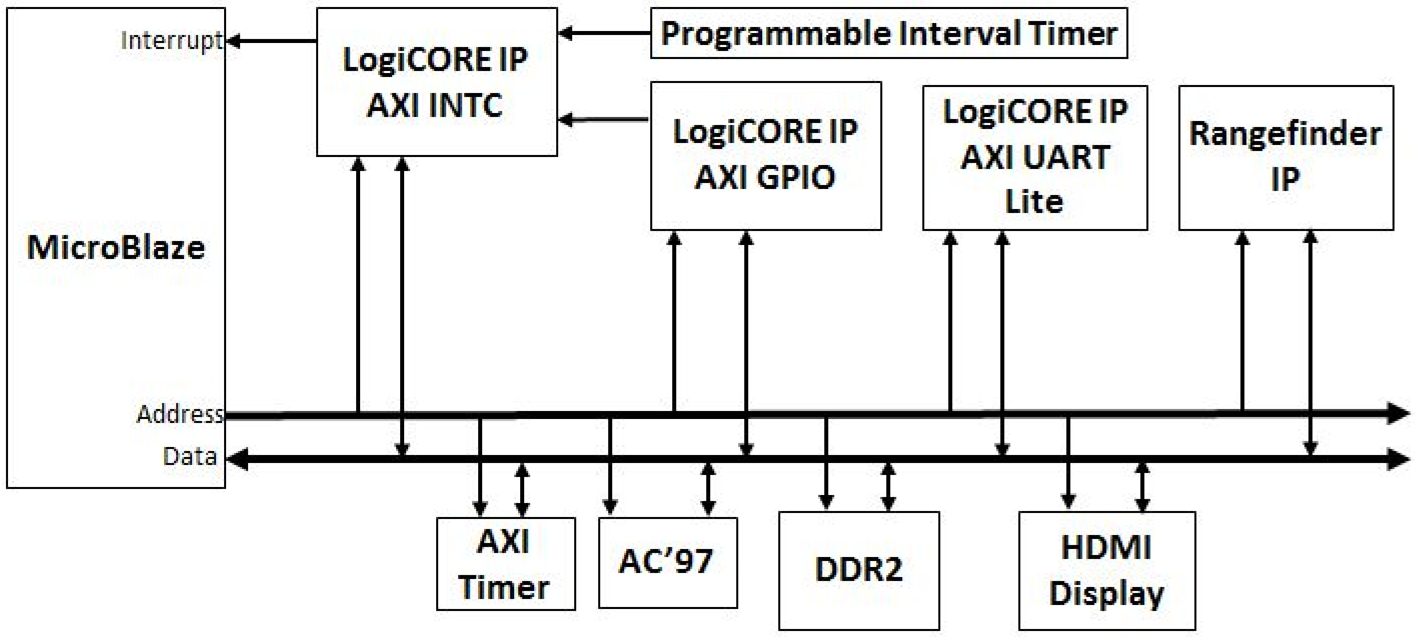

Figure 6: System Block Diagram

In order to physically interface the ATLYS board with the HC-SR04, the Digilent VmodBB was connected to the ATLYS board's VHDCI connector. Then, the following three pins were added to the system's UCF.

```
NET rangefinder_0_echoIn_pin  LOC = "V16"   |IOSTANDARD = "LVCMOS33"; # Sch name = EXP-IO1_N
NET rangefinder_0_trigger_pin LOC = "V15"   |IOSTANDARD = "LVCMOS33"; # Sch name = EXP-IO2_N
NET rangefinder_0_photoIn_pin LOC = "V13"|IOSTANDARD = "LVCMOS33"; # Sch name = EXP-IO3_N
```

The actual setup is shown in Figure 7.

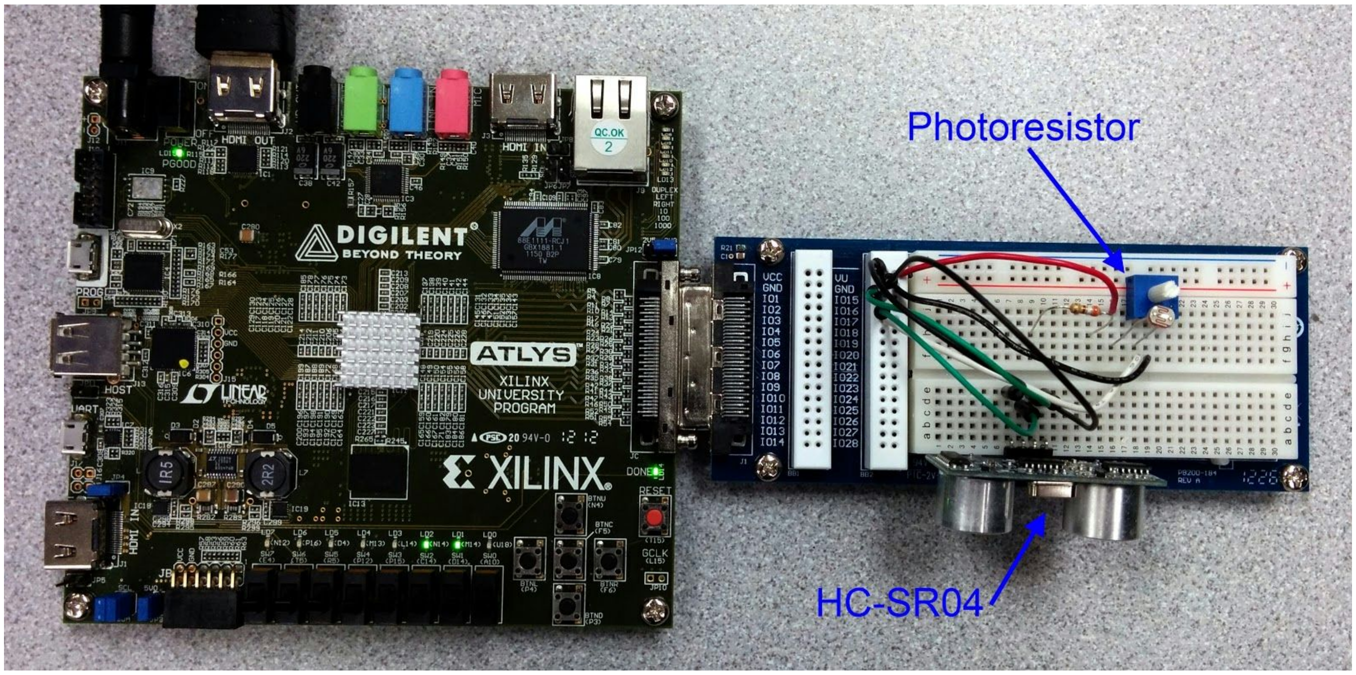

Figure 7: Physical System Layout

Bug Report

BUG	SYMPTOM	SOLUTION
UCF File	The pins we thought were hooked up to the Rangefinder IP were not operating correctly.	The UCF references were just unclear. The second bank of pins on the VmodBB corresponded to the *_N pin names on the schematics. This meant that pin EXP-I01_N actually corresponded with pin IO15 on the board, and pin EXP-I02_N corresponded with IO16. Making this change fixed the problem.