University of Florida
Department of Electrical and Computer Engineering
EEL 5666
Intelligent Machines Design Laboratory

BallMan
Sensor Report

03/16/05
Vivek Manoharan

INTRODUCTION

Ball Man's purpose is to retrieve stray tennis balls on a tennis court and properly drop them off at a predetermined location. This requires a sequence of behaviors from Ball Man prompted by his sensors. A bump sensor is used as an initialization key to know what color tennis ball is being used; complementary to that, a photocell resistor is used to determine the lighting conditions and how the tolerances should be changed based on that certain lighting.

After that is completed, it scans the tennis court for any stationary tennis balls that are lying on the ground, using a CMU camera. Upon detecting a tennis ball, the robot moves full speed ahead until its ultrasound array detects an object. Provided that the CMU is still tracking the tennis ball, the robot will slow down and adjust itself so that it is centered upon the tennis ball. Any difference in ultrasound pings between the two sensors will cause one of the motors to slow down to allow Ball Man to "turn" in the proper direction.

Once the pings show that the tennis ball is approximately 4 inches away from Ball Man, the servo-controlled arm drops down and uses Velcro to pick up the tennis ball. Using a screw-like mechanism, the tennis ball can will rotate to pry the tennis ball off of the Velcro, and allow it to fall into the can. Once this has happened two more times, the CMU will try and locate a bright red flag (on the tennis court fence) and approach it with high speed. Once within 4 feet of it, it will stop, release the tennis balls, and resume finding more tennis balls.

Bump Switch

Instead of being used as a bump “sensor”, it’s being used as an “ENTER” key for Ball Man. This is necessary for getting the base RGB values of the tennis ball and initializing Ball Man for the specific lighting levels. Also, it allows for Ball Man to be stopped after it has found three tennis balls. This button has a multitude of purposes, For now, this switch is being set up on PC3. This is the same port as the LCD port, but PC3 is not connected to anything. Therefore, it is a very useful digital pin to use and not intrude upon the other ports.

The bump switch is active high, meaning that it requires a ‘1’ to continue through with some functions. i.e.

```
while( ! enter_key());
```

So, the program is stuck until the user hits this switch. PC3 is pulled down to GND the whole while until the switch allows it to become ground. `enter_key()` returns a ‘0’ with the absence of the push and a ‘1’ when the button is recognized.

The bump switch is not a very complex switch. It was free and was obtained in lab.

Photocell Resistor

This resistor is used to calculate how much ambient light is present in order to assist the CMU camera during its calibration routine. Though it seems like a simple enough sensor, it is very crucial to know what range of values to accept after the color initialization sequence begins. Depending on the lighting outside, it can cause the tracking mechanism to have a standard deviation of anywhere from 10-100 for the individual RGB values.

The “light-sensor” is set to PF0. This analog port needs to be used to assess the value that the photo-resistor is giving to Ball Man.

Ultrasound Array

Two SRF-04 ultrasound rangers are being used for this array. Both were purchased from Acroname. They are both set at 70° off of the horizontal. The purpose of the rangers is to enable Ball Man to detect when something is close by (since it will be traveling at a relatively velocity), and if it in fact matches with the CMU camera's tracking command, it will approach the "tennis ball".

The ultrasound devices are not fired at the same time. This prevents any chance of one sensor receiving the other's data. They will be firing one after the other, allowing for the 32 ms timeout ping that will occur at first when Ball Man is facing a giant empty court.

Once one sensor picks up some data other than timeout, compare mode starts. This is where the pulse distances are compared between the two ultrasound sensors. If for example the right sensor is reading a shorter pulse than the left sensor, this means that the object is closer to the right sensor. Therefore, the right motor should slow down and allow Ball Man to sway slightly to the right gradually with time. Once the times are equal, both speeds will be equal again and the ball should be dead center ahead.

PB0 and PB1 were used as the Pulse output and the Echo input (respectively) for the first ultrasound sensor. PB2 and PB3 were used in a similar fashion for the second ultrasound sensor. A single timer (Timer 0) was used for both calculations since neither pulse was being counted simultaneously.

Ultrasound code snippets can be found in the Appendix.

CMU Camera – SPECIAL SENSOR

The CMU camera is being used to detect not only the tennis ball, but also the home base. The tennis ball will be a typical “just-shy-of-florescent-green” color, while the home base will be a bright red. These two contrast very well on a tennis court, and can therefore be spotted out by the CMU camera in most lighting situations. The CMU camera operates through the USART0 from the Atmega128. PE0 and PE1 are used for the USART Rx and Tx respectively. Code was borrowed from Anthony Huereca, Spring 2004 in order to get the USART working.

The CMU camera uses middle mass mode and tracking color mode to locate certain color values. As long as coordinates are fed back through the Rx line of the USART, the motor will continue moving forward. Otherwise, the camera pivots in place; one motor moves forward at 1/10 full speed, and the other moves backward at 1/10 full speed.

The CMU camera was purchased from Acroname, and required assembly. Though it wasn’t very difficult, it was time consuming, and proper joints were imperative for proper CMU functioning.

EXPERIMENTS

BUMP SWITCH

To test the bump switch, data was collected before the switch was hit, during the push, and after the push. The switch worked well, since it was only working through a digital I/O pin (PC3).

A test program was written where nothing would be written to the LCD screen until the bump switch was pushed. This allowed me to ensure that my enter key was working properly, and that I could have a confirmation button for many purposes.

ULTRASOUND RANGERS

Experiments were conducted with a single and a dual configuration.

EXPERIMENT 1

The single configuration matched the accuracy of the pulse with the actual distance of the object. Measurements were taken with a yardstick and from the end of the SRF-04 module's "speaker" for the "Actual" column. Timer0 and PB1:0 were used for this test. The following equation was used for computing distance from the pulse data:

```
Result=TCNT0*1.;    //makes Result a float
delay_ms(10);       //Mandatory Delay
return ((Result + Count*256) * 0.432);
```

count contains data of an overflow, and 0.432 is the ratio of 64/148 (this is obtained from the fact that 1024 pre-scaling is used and that 1/148 us = 1 in.

. The following data was obtained:

Trial	Measured	Actual
1	1.1280"	1.0"
2	3.0900"	3.0"
3	5.1150"	5.0"
4	10.1928"	10.0"
5	15.3250"	15.0"
6	21.0547"	20.0"
7	30.6853"	30.0"

8	41.3278”	40.0”
9	62.1732”	60.0”
10	93.5268”	90.0”

Four decimal point precision was used by converting the integer values to float values, and outputting that to an LCD screen. Also, make sure you conduct tests in an extremely open area. Since the SRF-04’s send out a conical wave, a lot can be hit. Trial 10 gave some trouble because of this; other items were send back waves.

Overall, this experiment proved that the accuracy of the sonar ranger is pretty strong. As long as the SRF-04 gets a reading for far distances, like the 90.0” trial, that’s good enough for Ball Man. Finally, sending out float values to the LCD screen can be very frustrating. Sometimes it works, most of the time it doesn’t.

EXPERIMENT TWO

In the second experiment, the difference between the two rangers is tested. One ranger is set up on PB1:0, and the second is setup on PB3:2. I measured the difference as measured by a yardstick to what was needed to get “equal values” between the two sonar sensors. I tried to approximate similar distance measurements as in experiment one, but obviously some of the larger distances were not possible.

Trial	Difference between sensors (in.)
1	0.125”
2	0.068”
3	0.125”

4	0.313”
5	0.375”
6	0.563”
7	0.438”
8	0.438”
9	0.563”
10	0.625”

Three decimal point floats were used. This experiment proved that there is not a significant difference between the two rangers. Furthermore, difference calculations for motor control will be precise.

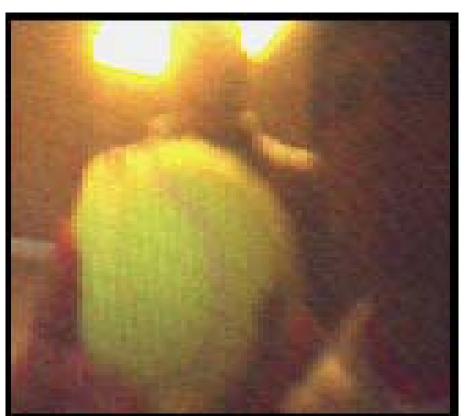
CMU Camera

The CMU camera experiments include detecting and giving proper x-y coordinates based on the objects position in its screen. The three experiments include noting the effects of white balance on screen dumps indoors, noting the effect of white balance on screen dumps outdoors, and the ability to distinguish between a tennis ball green and a bright red color outdoors.

EXPERIMENT

With white balance off indoors:

With white balance on indoors:



As you can see, the “fluorescence” of a

tennis ball is more apparent when white balance is on. The light in the upper part of the image is the light in my room, and therefore indicates natural sunlight outside should provide an accurate depiction of the tennis ball. The light in my room is a fluorescent bulb.

This test was conducted using the java program that can be obtained on the Carnegie Mellon website.

More experiments were conducted once the TW command was discovered. This command enables the user to set an object in front of the CMU camera and after it takes a brief snapshot of the center mean color values, it will track that color. These tests came out extremely well. The only problem was the contrast of the tennis ball on surfaces other than a tennis court. Since the color values almost produce a brownish hue, many floors and indoor items can trigger the CMU camera. Even if Ball Man is near grass and some dirt, that could trigger the tracking color command. Therefore, use during nighttime is much better than daytime (eliminates background colors), and indoor use is not suggested.

CMU CODE:

The following code shows how my interrupt works to track the color that I set it to receive, and as long as the M packet that is sent back has an x and a y value, then that means that the CMU Camera has found your desired object.

```
SIGNAL(SIG_UART0_RECV)
{
	//get data
```

```
if(recpos == 900)
{
	recpos = 0;
	readpos = 0;
}
unsigned int foo=UDR0;

if(uart_data[(recpos-2)] == 77)
{
	mx = uart_data[(recpos-1)];
	my = foo;
	if(s)
	{
		if(mx || found)
		{
			if(!found)
			{
				//cli();
				motora = 0x00;
				motorb = 0x00;
				lcd_clear();
				lcd_puts("found object!");
				delay_ms(2000);
				found = 1;

			}
			else
			{
				if(mx == 0)
					mx = mx_backup;
				if(my == 0)
					my = my_backup;

				if(ball_count == max_balls)
					line_up_cone();
				else
					line_up_ball();
			}

		}
	}
	else
		lcd_clear();
}

uart_data[recpos++]=foo;}
```