Direction Control of Robotic Fish Using Infrared Sensor Modules and IPMC Activation Schemes with a dsPIC30F4013 DSC

Carl A. Coppola
04/03/2009

ECE 480, Team 04
ME 481, Team 09
Abstract

This application note discusses one particular method of obstacle avoidance for an autonomous robotic fish propelled by an IPMC fin, using a series of analog output infrared (IR) modules for object distance detection. Using the multiple IR modules allows for a more precise closed-loop control, which will allow for better, less obstacle-interrupted movement of the fish.

keywords: IPMC, IR sensor, analog input, A/D conversion
## Table of Contents

Introduction ....................................................................................................................1  
Objective ........................................................................................................................1  
Issues ..............................................................................................................................1  
Example .........................................................................................................................5  
Results ............................................................................................................................6  
Conclusions ....................................................................................................................8  
References ......................................................................................................................8  
Appendices ..................................................................................................................9    
  Appendix A: dsPIC03F4013 .........................................................................................9  
  Appendix B: 10-bit ADC for Multiple Analog Inputs ..................................................10  
  Appendix C: Robotic Fish IPMC Input Waveforms ......................................................11  
  Appendix D: Sharp IR Module GP2D120XJ00F ..........................................................12
**Introduction**

Key to the proper function of any autonomous object is the ability to both detect and subsequently avoid obstacles. Object detection can be utilized by various different methods, however, application and environment are the most important factors in determining which method to incorporate. In the case of the robotic fish, slow movement in water, lengthy turns, and low power requirements resulted in a series of three IR sensor modules to be chosen in order to provide a more robust detection of obstacles. The IR modules are an inexpensive, durable, lightweight, and relatively accurate method of object detection that requires little power and can be implemented with little difficulty, both mechanically and electrically.

**Objective**

The objective of this note is to describe how to convert the analog output voltages from the IR sensor modules to digital signals so as to determine which IPMC scheme to enable in order to turn the fish right or left, or keep it moving straight.

**Issues**

When using IR sensors in object detection, there are multiple issues which arise, first and foremost of which being the detection range of the sensor to be used. In order to detect further out from the robot, the intensity and sensitivity of the emitter and receiver must increase, so that the signal does not dissipate as rapidly as it normally would. This generally increases the size and/or weight of the module. Likewise, to see very close to the robot, the light must be focused and the sensitivity and noise filtering must also increase for the module to be able to detect the minutely small difference in phase between emitted and received light frequencies. All of these factors increase the cost and power requirements of both types of modules. In addition, modules with broad detection ranges tend to be both larger in size/weight and power consumption, as well as much more costly. Thus, one must balance size/weight, power consumption, cost, and detection range together when choosing which model sensor to buy.

In addition to the detection range of the sensor(s) to be used, the number of sensors required is of great importance. While most small and/or nimble land-based robots using servo- or dc-motor-driven wheels can adequately use two sensors to detect and avoid objects with little to no trouble, slow-moving aquatic-based robots with lengthy response times, such as the IPMC-driven robotic fish, require at least 3 IR sensor modules to detect and respond to objects in their vicinity. This is due mainly in that using two sensors requires that they be at equal angles from the central axis of the robot, and that the larger the angle, the closer to the obstacle the robot will be before detecting it. On the other hand, making the angle between them smaller increases the risk of running into something on either side because the angle of incidence is too large to be adequately reflected back to the sensor, if it is reflected at all. Figure 1 briefly shows the robustness of a 3-module scheme.
Another important issue is one of converting the output of a sensor to something that the controller being used can recognize. There are two basic types of output signals: digital and analog. Digital signals require encoding and, with the use of the proper codec, decoding in order for the controller to process the signal. If the module is already compatible with and/or made for the controller being used, then a module with digital output would most likely be much easier than integrating a module with analog output, since the analog output may need to run through an analog-to-digital converter (ADC) before being sent to the controller, depending on what type of controller is being used. If the module is not directly compatible with and/or made for the controller being used, then an analog output is most likely the best way to go. In the case of the robotic fish, we are using a digital signal controller (DSC) and IR sensor modules with analog outputs. There is an example of our A/D conversion code for using only one IR module in the next section.

In much the same way that A/D conversion is an important issue, simultaneous A/D conversion of multiple analog inputs is as much an, if not more of an important issue. Though an example from [1] for a 10-bit ADC is included in Appendix B, simultaneous A/D conversion requires the use of buffering in order for the controller to be able to process the analog signals being output from each of the devices. This is important in that using three IR modules with analog inputs requires such A/D conversion in order to detect and avoid obstacles in the robot’s periphery.

The last, but by no means least important issue is using the input signal from the modules to determine which direction to move the robot. Doing this requires use of a closed-loop control and three general IPMC schemes. These schemes, shown in Figure 2, would be left-turn, straight, and right-turn, respectively.
Looking back at Figure 1, the two left-most situations would activate the right-turn scheme, the center situation would activate either the left-turn or right-turn schemes, and the two right-most situations would activate the left-turn scheme. Figure 3, shown below, presents the current IPMC activation code for the straight scheme. It uses either simple square waves or pulse-width modulation (PWM) to actuate the IPMC fin. In order for the left-turn or right-turn schemes to actuate properly, square waves are used with no PWM ([3], App. C), as can be seen in Figure 4.
Figure 4. IPMC actuation code for right-turn and left-turn schemes. Uses simple square waves to keep the voltage oscillating on only one side of the IPMC.

Integrating these coding schemes into the existing code could be accomplished through the use of conditional statements, such as “Select…Case” statements or “If…Then” statements. For example, if the front and left IR modules both sense something within a certain critical distance, then a conditional statement resembling the one shown below,

```c
if ((IRFrnt==1)&(IRLeft==1))
{
    turn_angle=0;
    // Either insert right-turn actuation code itself, or call existing actuation code.
}
```

would initiate the right-turn actuation scheme, thus preventing the fish from running into the aforementioned wall.
Example

This section goes over the code through which the analog signal output from a single IR module is converted to a digital signal by the 12-bit ADC in our DSC, the dsPIC30F4013 ([2], App. A). The figure below (Figure 5) shows the current code for the IR module.

![Code snippet](image)

Running through the code line-by-line, the “#include<...>” statement defines the header file(s) to be used; in this case, the header containing the codec and function list for the dsPIC30F4013 DSC. Next, the two variables “IRsensor1” and “counter” are defined as integer-type variables. The main function is then defined. As can be seen by the comments in the code above, the next 9 lines both initialize and set the states of the necessary registers for the A/D conversion that follows. The “while (1)” conditional statement causes the code within itself to repeat continuously. This “nested” code first starts the sampling by setting the SAMP bit in the “on” state of 1, after which it is set back to the “off” state of 0, so that conversion can begin. The next “while” statement loops until the DONE bit of the ADCON1 register is in the “on” state, signifying that the conversion is done. The last segment of the code writes the value of the converted signal to the IRsensor1 variable. Once this has been done, the sampling begins again.
Results

To test if the code for the single IR module analog output A/D conversion worked, a test circuit was constructed on a protoboard, then connected to a power supply outputting 5V DC. The output terminal was connected, in parallel, to both the AN9 terminal on the board and a digital multimeter so that the analog output voltage could be checked and compared with the digital value being read in the MPLAB IDE program. Distance was gauged with a small 15cm ruler, and a piece of white paper was used for the reflective surface. First, the voltage read by the multimeter was compared with the spec voltage at the same distance from sensor ([4], App. D). Once everything was determined to be correct, the converted signal was compared to the reading from the multimeter to ensure that the signal was being encoded correctly. This was done using a reference voltage of 5V and the 12-bit maximum resolution value of $2^{12} = 4096$ as computed in equation (1).

$$V_{converted} = \left( \frac{5V}{4096} \right) (DigitalValue) \tag{1}$$

The following figures show two different distances and their corresponding voltage values, as read from the multimeter, ruler, and MPLAB IDE watch.

Figure 6. Distance 1: approx. 8.5cm, should have voltage just above 1.4V DC.

Figure 7. Voltage read by multimeter at distance 1.
Using the value given in Figure 8 and plugging it into eq. (1), the voltage is calculated to be $1.427\text{V}_\text{DC}$, which is approximately the same value as what was read by the multimeter.

Figure 9. Distance 2: approx. 12.5cm (as paper was leaning forward), should have voltage just above $1.0\text{V}_\text{DC}$

Figure 10. Voltage read by multimeter at distance 2.
Using the value given in Figure 11 and plugging it into eq. (1), the voltage is calculated to be 1.080V\textsubscript{DC}, which is approximately the same value being read by the multimeter.

### Conclusions

Using analog IR modules to detect distance from obstacles is a relatively cheap, low-power, and reliable way to create a closed-loop control system for object avoidance. By setting up a control scheme such that certain module outputs define particular IMPC actuation schemes, a robotic fish can be set well on its way to being both purely autonomous and robustly functional. In doing so, these bio-mimetic automatons could become an efficient and low-maintenance solution to detecting, and possibly preventing, several environmental issues present today.

### References


Appendices

Appendix A: dsPIC03F4013

Figure #. Pin diagram for dsPIC30F4013 DSC [2].

Figure #. Block diagram for dsPIC30F4013 DSC [2].
Appendix B: 10-bit ADC for Multiple Analog Inputs

Figure #. 10-bit ADC scheme for 4 analog inputs with simultaneous sampling [1].

```
ADPCFG = 0xFF78;  // RB0, RB1, RB2 & RB7 = analog
ADCON1 = 0x000C;  // SIMSAM bit = 1 implies ...
                  // simultaneous sampling
                  // ASAM = 1 for auto sample after convert
                  // SSRC = 111 for 3Tad sample time
ADCHS = 0x0007;   // Connect AN7 as CH0 input
ADCSS1 = 0;       // Auto Sampling 3 Tad, Tad = internal 2 Tcy
ADCON3 = 0x0302;  // CHDS = 1x implies simultaneous ...
                  // sample CH0 to CH3
                  // SMPI = 0011 for interrupt after 4 converts
ADCON1bits.ADON = 1; // turn ADC ON
while (1)            // repeat continuously
{
    ADC16Ptr = &ADCBUF0;  // initialize ADCBUF pointer
    OutDataPtr = &OutData[0]; // point to first TXbuffer value
    IPS0bits.ADIF = 0;   // clear interrupt
    while (IPSObits.ADIF);  // conversion done?
    for (count = 0; count < 4; count++) // save the ADC values
    {
        ADCValue = *ADC16Ptr++;
        LoadADC(ADCValue);
    }
}  // repeat
```

Figure #. 10-bit A/D conversion code for 4 analog inputs with simultaneous sampling [1].
Appendix C: Robotic Fish IPMC Input Waveforms

Figure #. IPMC control signals generated onboard robotic fish [3]. From top, (a) square wave, (b) pulse-width modulated square wave, (c) sinusoidal signal.
Appendix D: Sharp IR Module GP2D120XJ00F

Figure #. Analog voltage vs. distance calibration curve [4].
Figure #. Physical dimensions of IR module [4].
Figure #. Circuit schematic of the IR module[4].