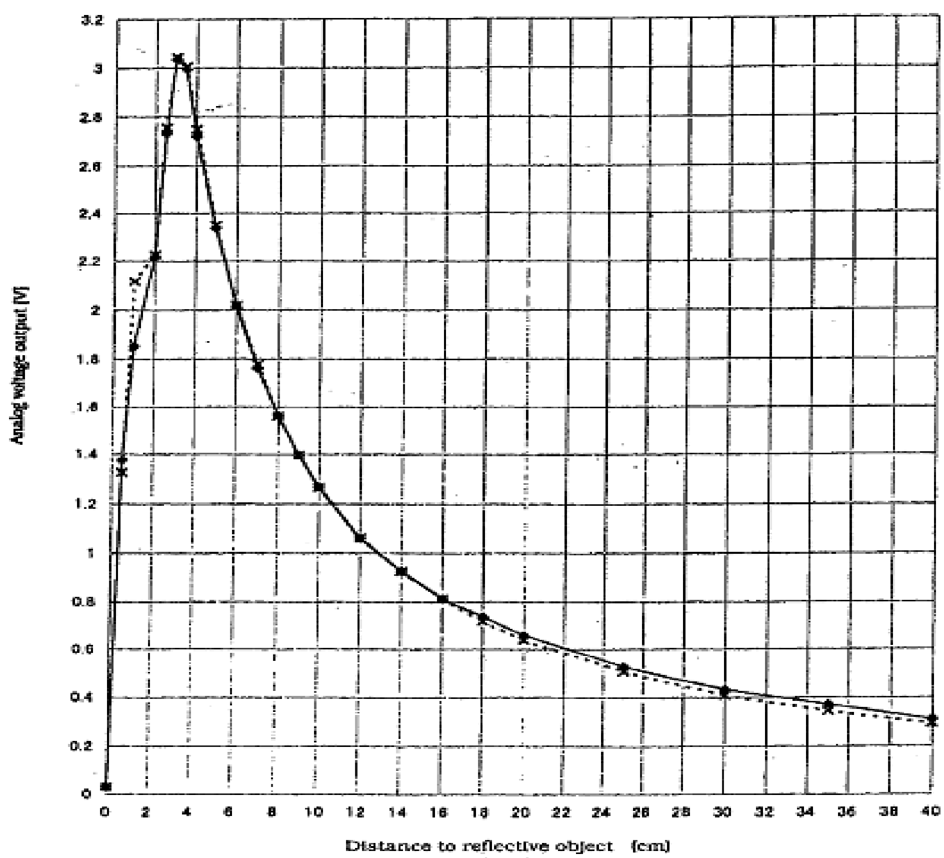
Embedded Hardware Design

Experiment 4/5 : Path Planning using IR Sharp Sensor 15 Sept. 2010

In this experiment, an IR Sharp Sensor will be interfaced with the microcontroller to measure the distance of any object and to send the distance data for a moving object to the host computer for displaying the distance profile through MATLAB.

An IR Sharp sensor consists of an infrared transmitter and an array of infrared detectors matched to the transmitter wavelength. The transmitter sends out infrared waves, and the detectors sense the waves reflected by objects in front of the sensor. The sensor contains signal processing circuitry which use triangulation method to find out the distance of the object and provide an analog output corresponding to the distance of the object. The update rate of Sharp sensors is about 25Hz and noise levels are around 200mV.



I. Distance Measurement using the IR Sharp Sensor

1) Connect Vcc and Gnd to IR sharp sensor pin 1 and 2. Connect a 22µF bypass capacitor between these pins, observing proper polarity, as near the device as possible.
2) Connect the Sharp sensor output (pin 3) to one of the microcontroller ADC inputs (say PA0).
3) Write a program that will generate the ADC output in ADC Single Conversion Mode. (Relevant Registers: ADMUX, ADCSRA, ADC).
4) Make a look up table corresponding to the given plot of sensor output vs. distance and use it to compute the actual distance corresponding to the ADC output.
5) Write a program to print the ADC output and the computed value of the distance on the LCD. Measure the actual distance manually and compare the two.

II. Distance Profiling of a Moving Object

1) Keep the same connections as done in part A. Only remove the LCD from your boot loader kit.
2) Connect the USART output pins to a RS232-to-USB adapter, which connects the microcontroller to the USB port of the host computer. The USB adapter has four pins. Connect the black wire of the adapter to ground of the microcontroller kit, the red wire to the Tx pin of the microcontroller i.e. PD1 and the orange wire to the Rx pin of microcontroller i.e. PD0. Leave the brown wire with no connection (NC).
3) Write a routine to initialize the USART. Set the baud rate to 57600 and enable Tx.
4) Write a routine USART_TransmitByte() to transmit a byte serially. As the 10-bit ADC output cannot be put in a single frame, one has to create two frames, each containing 5 bits of data. In order to distinguish the two 5-bit portions, a header has to be included, constituting three bits on the MSB side in the frame. Add a header 000 for the lower half of the data and 100 for the upper half.
5) Send the 10 bit ADC data though USART, requiring a total of 20 bits to be sent. (Relevant Registers: UBRRL,UCSRA,UCSRB,UDR).
6) Receive the ADC data from microcontroller on MATLAB, using the MATLAB program provided to you. MATLAB will parse the USART data and plot those values. It will also write the ADC data in a file “data.txt” in tabulated form in two columns. First column corresponds to ADC data and second one to time stamp of that sample.

Note: Given that ADC conversion takes 13 clock cycles, and that the microcontroller clock frequency is 12 MHz, the microcontroller takes about one microsecond to perform an ADC conversion. Moreover, the USART will require a serial data transmission time, given approximately by 0.4 msec (verify). Compared to these delays, a Sharp sensor takes around 40 ms to do update its output. A suitable delay should therefore be introduced after every data transmission to prevent transmission of redundant data.