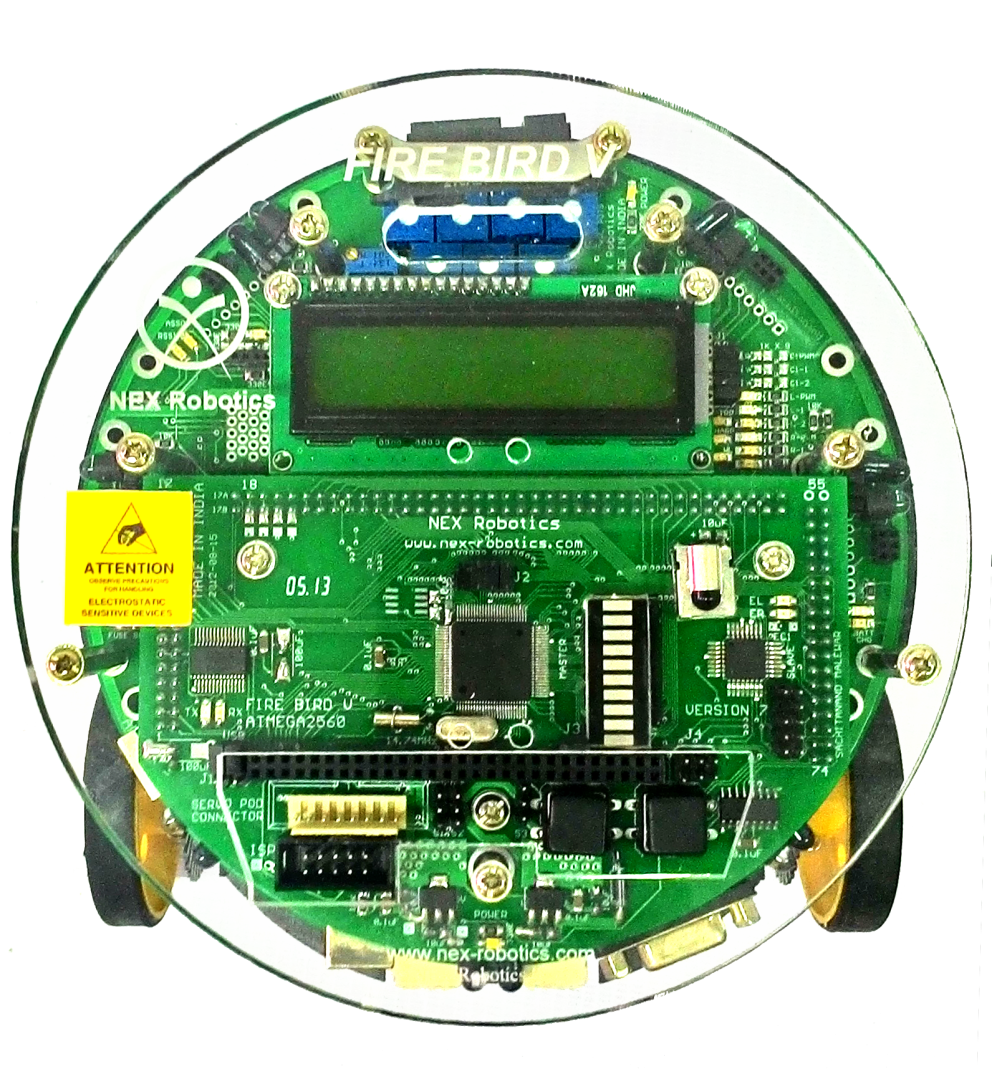
**Lab Experiments**

This document is meant to be used by teachers in the following departments:

**Electronics, Electronics and Telecommunication, Computer Science, Mechanical Engineering, and Information Technology**

****

**Index**

|  |  |  |
| --- | --- | --- |
| Serial No. | **Details** | **Page No** |
| 1 | Introduction | 3 |
| 2 | Basics of Microcontroller | 4 |
| 3 | Basics of Fire Bird V |  |
| 4 | IO Interfacing | 6 |
|  | LCD Interfacing | 10 |
|  | Position and Velocity Sensor | 15 |
|  | Motor Interfacing |  |
|  | Pulse Width Modulation |  |
|  | Interrupts |  |
|  | Interfacing GPS Module |  |
|  | Wireless Communication – ZigBee |  |
|  | USB to Serial Communication |  |
|  | Interfacing Analog-Digital Conversion |  |
|  | Short Theme/Theme activity |  |

**Introduction**

Dear Teacher,

Greetings from e-Yantra!

The lab experiments presented in this document are meant to help you, to incorporate hands-on training for your students by using the robots and accessories from your e-Yantra lab. We have tried to map these experiments to topics you teach as part of your course curriculum. However, a successful training is one which is locally customized according to the requirement. Please maintain your style and feel free to extend these exercises to get the best out of your students.

We hope to hear from you about your experiences and improvisations!

e-Yantra Team

**Basics of Fire Bird V**

Fire Bird V is designed by NEX Robotics and Embedded Real-Time Systems (ERTS) lab, Department of Computer Science and Engineering, IIT Bombay. Its modular architecture, allows you to control it using multiple processors such as 8051, AVR and ARM7. Different family of microcontrollers can be added by simply changing the top microcontroller adaptor board. This feature makes Fire Bird V robots very versatile.

You can watch the video tutorial present in the given folder on how to change the adaptor board.

Currently we are using P89V51RD2 (8051 - AVR) microcontroller adaptor board. The Fire Bird V components required for learning the concepts covered in this handbook are:

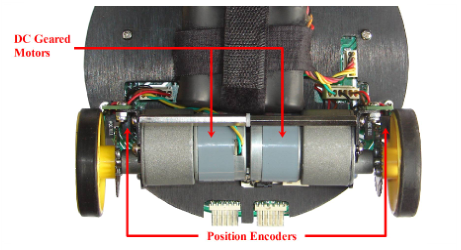
1. LCD:



**LCD**

**Note: You can refer to LCD module from e-Yantra DVD tutorial in hardware and software manual.**

1. Position Encoder:



**Figure 1.1: Placement of Position Encoder on Fire Bird V**

**Note: You can refer to Robot Position Control module in hardware and software manual.**

**Topic:** To study Input-Output Interfacing

**Prerequisites:** C Programming, Basics of Embedded C

**Software Manual:** IO interfacing (Section 3)

**Components Required:** Fire Bird V Robot with 8051 Adapter Board

**Lecture Notes:**

What does input and output device mean?

Input devices provide input to the computer/microcontroller, while output devices provide a way for a computer/microcontroller to output data for communication with users or other device.

For e.g.:

Input devices: Keyboard, Switch, Sensor

Output devices: LCD, Motor, LED

How do we connect the input and output devices?

Input/output ports are the connections available in microcontroller for to connect input/output devices.

Ports are junctions to connect the input/output devices. There are different types of ports.

One is GPIO (general purpose input output), which is an interface available on modern microcontrollers to provide an ease of access to the devices internal properties.

**Interfacing Input-Output device:**

There are total 40 pins in 8051 microcontroller, where 32 pins are used for input and output.

Pins are grouped together and are called as Ports. There are total four 8-bit ports, that is, Port 0, Port 1, Port 2 and Port 3.

Port 0 is a dual purpose port, it is located from pin 32 to pin 39.

Port 1 is a dedicated I/O port from pin 1 to pin 8.Upon reset it is configured as output.

Port 2 is a dual-purpose port, located from pin 21 to pin 28.

Port 3 is also dual-purpose port, located from pin 10 to pin 17.

A logical state on a pin determines whether it is configured as input or output. If a pin on microcontroller needs to be configured as output, then logic 0 should be applied to the appropriate bit on I/O port. So, a voltage level on the appropriate pin will be 0.

Similarly, if a pin needs to be configured as input then, logic 1 should be applied to the appropriate port. Here, the voltage level on appropriate pin will be 5V.

Let us see few examples on working with ports:

Example 1: Set the 4th pin of Port 0 (i.e. P0^3) to logic 1

**Solution:**

{

P0^3=1;

**While** (1);

}

Example 2: Set Port 0 as an Input port.

**Solution:**

{

P0=0xFF;

**While (**1**);**

}

You can refer to hardware manual to know the port connection that is, the pin configuration for individual device in Fire Bird V. Figure 1.1 shows the basic pin configuration.

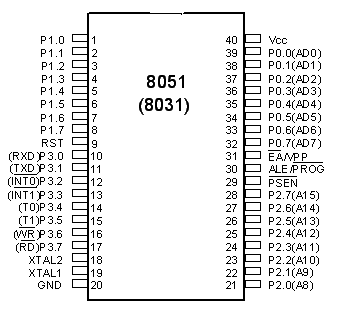


Figure 1.1: 8051 pin configuration

**Problem Statement:** Program the robot to buzz the buzzer for 1 second.

**Interfacing buzzer in 8051:**

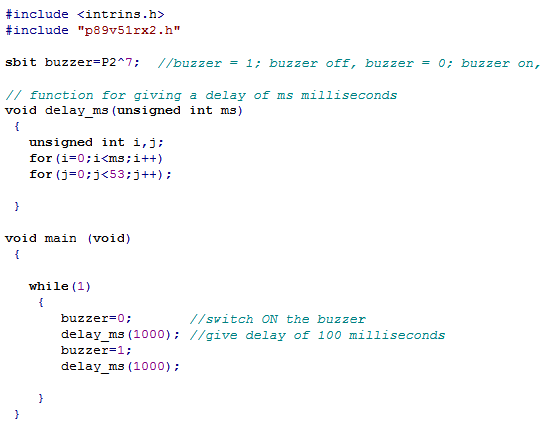
In Fire Bird V, buzzer is situated on the main board and connected to the 7th pin of Port2.

To turn buzzer off apply logic 1 to P2^7 and to turn it on we have to set P2^7 to zero.

**Algorithm:**

1. Start
2. Assign the port value to buzzer
3. Make buzzer on for 1 second
4. Stop buzzer
5. End

**Program Snippet:**



Here, in above code

#include directive is used for including header files in the code.

sbit data type defines a bit within special function register(sfr), which is defined as

sbit *name* = *sfr-name* ^ *bit-position*;

In our code,

sbit buzzer = P2^7;

where, name is buzzer, special function register is P2 and bit position is 7

The function delay\_ms() provides a delay of milliseconds where ms is the value passed to it by the calling function

In the function main(), the while loop toggles the buzzer ON and OFF with a finite delay of 1000 milliseconds.

**Loading program on Fire Bird V:**

To load the program on Fire Bird V robot, follow the instructions given in software manual chapter 2.

You can burn the code on Fire Bird V using USB to Serial converter.

**Extension:**

Program the robot to play Nokia SMS tone.

Program the robot to blow buzzer when switch is pressed. When switch is released it should turn off.

**Evaluation/Conclusion:**

**Topic:** To study LCD Interfacing

**Prerequisites:** I/O programming

**Module covered from software manual:** LCD Interfacing (section 6)

**Components Required:** Fire Bird V Robot with 8051 Adapter Board

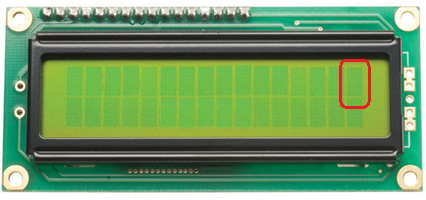
**Lecture Notes:**

* **Ask the students, what is LCD?**

Liquid Crystal Display (LCD) is a flat panel used for displaying information such as text, number and special characters.

* Tell the students that in Fire Bird V, 16x2 Alpha Numeric LCDis used shown in Figure 2.1.
* A 16x2 LCD can display 16 characters per line and there are 2 such lines. In this LCD each character is displayed in 5x7 pixel matrix as shown in Figure 2.2.

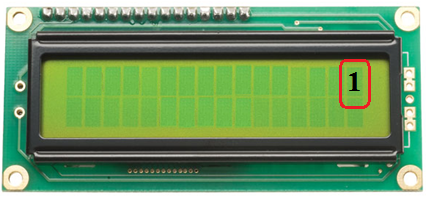
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |



5x7 pixel matrix

Figure 2.1: 16x2 Alphanumeric LCD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |



5x7 pixel matrix

Figure 2.2: 16x2 Alphanumeric LCD

**Pin Configuration:**

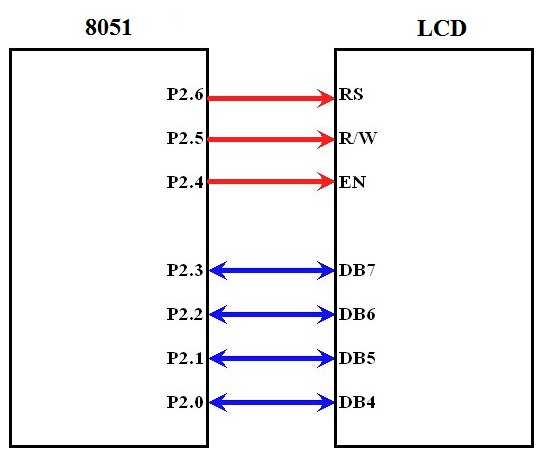


Figure 2.3: Pin Configuration

* 16x2 LCD has 16 pins, where:

|  |  |
| --- | --- |
| Pin | Description |
| Vss | Ground |
| Vdd | Supply Voltage |
| Vee | Contrast Voltage |
| RS | Register Select |
| Vss | Ground |
| RW | Read/Write |
| E | Enable |
| D0-D7 | Bidirectional Data Bus |
| Vdd,Vss | Back Light Supply |

**LCD Interfacing:**



Control lines:

These pins are used for controlling the way LCD works:

* Register Select:
  + - Register select control line is connected to P2^6.
    - There are two very important registers inside the LCD.
    - If RS =0, the instruction command code register is selected, allowing the user to send a command such as clear display, cursor at home, etc.
    - If RS = 1, the data register is selected, allowing the user to send data to be displayed on the LCD.
* Read/Write Select:
  + - Read/Write control line is connected to P2^5. This pin is used to select whether we want to write on the LCD or we want to read from it.
    - If RW is set a value of 0, it means we are writing to the LCD
    - If RW is set a value of 1, it means we are reading from the LCD.
* Enable pin:
  + - The enable pin is used by the LCD to latch information presented to its data pins.
    - When data is send to data pins, a high to low pulse must be applied to this pin in order for he LCD to latch the data present in the data pins.

Data lines:

Data lines are used to pass the data to LCD.

* There are total eight data pins named from D0 to D7
* Same eight data pins are used to send both, the **command** and the **data**.
* LCD is alpha numeric type and understands only ASCII format.
* We can interface LCD in either 8-bit mode or in 4-bit mode.
* In 8-bit mode, all eight data pins are connected with the microcontroller whereas in 4-bit mode only four data pins are connected to the microcontroller.
* In order to save the microcontroller pins, LCD is interfaced in 4-bit mode only.
* In firebird robot, LCD is interfaced in four-bit mode. So Data pin D4 to D7 is only used for interfacing. In this mode higher nibble and lower nibble of commands/data set needs to be sent separately.
* Busy flag:

To know the status of LCD, busy flag is used. The busy flag is D7 and can be read when R/W=1 and RS=0, as follows:

If R/W=1, RS=0. When D7=1(busy flag = 1), the LCD is busy taking care of internal operations and will not accept any new information.

If R/W=1, RS=0. When D7=0 (busy flag = 0), the LCD is ready to receive new information.

Important commands which is used for working with LCD:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Instruction | Hex | Decimal |
| 1 | Function Set: 8-bit, 1 Line, 5x7 Dots | 0x30 | 48 |
| 2 | Function Set: 8-bit, 2 Line, 5x7 Dots | 0x38 | 56 |
| 3 | Function Set: 4-bit, 1 Line, 5x7 Dots | 0x20 | 32 |
| 4 | Function Set: 4-bit, 2 Line, 5x7 Dots | 0x28 | 40 |
| 5 | Entry Mode | 0x06 | 6 |
| 6 | Display off Cursor off (clearing display without clearing DDRAM content) | 0x08 | 8 |
| 7 | Display on Cursor on | 0x0E | 14 |
| 8 | Display on Cursor off | 0x0C | 12 |
| 9 | Display on Cursor blinking | 0x0F | 15 |
| 10 | Shift entire display left | 0x18 | 24 |
| 11 | Shift entire display right | 0x1C | 30 |
| 12 | Move cursor left by one character | 0x10 | 16 |
| 13 | Move cursor right by one character | 0x14 | 20 |
| 14 | Clear Display (also clear DDRAM content) | 0x01 | 1 |
| 15 | Set DDRAM address or coursor position on display | 0x80 + address\* | 128 + address\* |
| 16 | Set CGRAM address or set pointer to CGRAM location | 0x40 + address\*\* | 64 + address\*\* |

**Problem Statement:** Display your name on LCD.

**Algorithm:**

1. Start
2. Send 0x28 as a command to initialize LCD in 4 bit mode.
3. Send 0x01 as a command to clear the display.
4. Send 0x0F as a command for cursor blinking.
5. Send data to be displayed on LCD.
6. Stop

**Extension:**

Write a program to display “e-Yantra” on 1st line and “Fire Bird V” on 2nd line.

Write a program to display special characters (@, &, \*, ☺, ☹, etc.).

**For more details, you can refer to 8051 Software manual section 6.**

**Evaluation/Conclusion:**

**Topic:** Study of Position and Velocity Sensors

**Prerequisites:** I/O programming, Motion Control Programming

**Module covered from software manual:** RobotPosition Control (section 4)

**Components Required:** Fire Bird V Robot with 8051 Adapter Board

**Lecture Notes:**

* **Ask the students, what are position and velocity sensors?**

Position sensors and velocity sensors are devices used to find position and velocity measurements respectively, that is, it can be used to find the relative position/displacement of the robot and the velocity with which the robot moves. In both, the common and most widely used sensor is position encoder also known as optical encoder.

* Tell the students that in Fire Bird V, position encoder **MOC 7811** is used.
* **How does this sensor work?**

This sensor has an IR LED and photo transistor, in front of each other separated by a slot and encased in black opaque casing as shown in Figure 3.1. An IR LED emits IR light and photo transistor detects the reflected IR light. The internal structure of the sensor is shown in Figure 3.2.

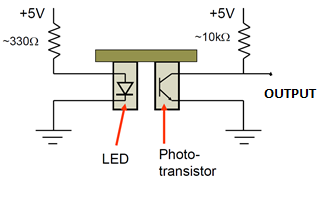
 ****

Figure 3.1 MOC 7811 Figure 3.2: Internal Structure of MOC 7811

When IR light falls on the photo transistor, the transistor turns on and acts as a closed switch, which gives minimum voltage at the output and is considered as logic 0. This is shown in Figure 3.3

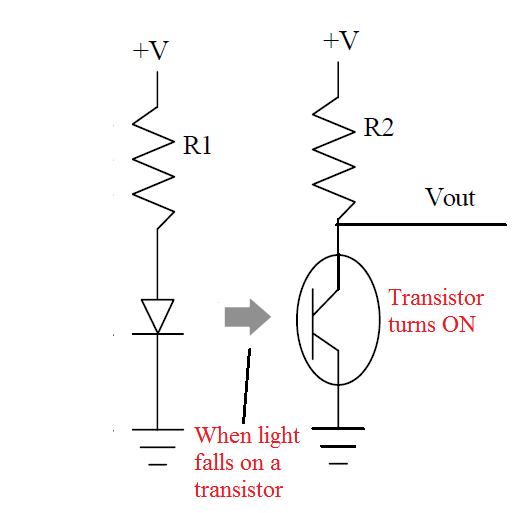
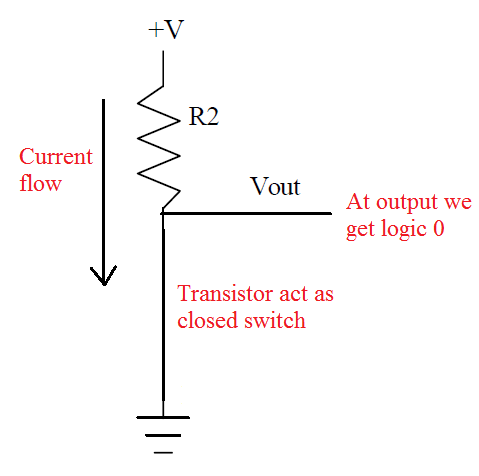
 

Figure 3.3 In the presence of light

Similarly, in absence of the IR light, the transistor turns off and acts as open switch, which gives maximum voltage at the output and is considered as logic 1. This is shown in Figure 3.4

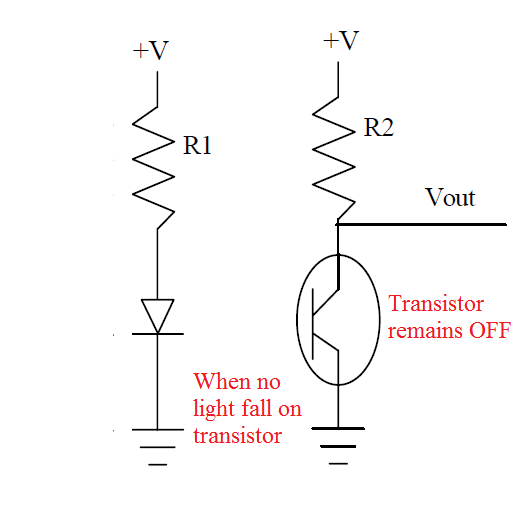
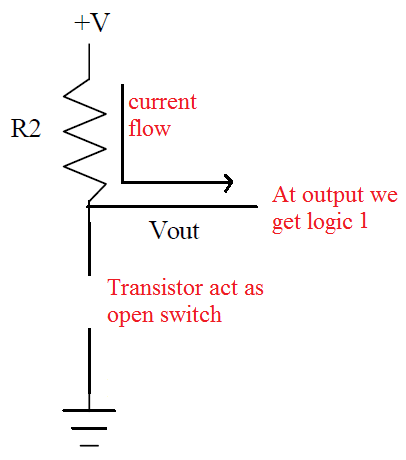
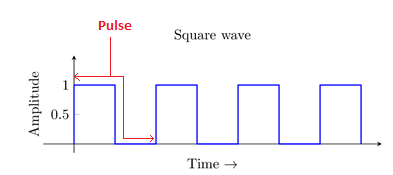
 

Figure 3.4 In the absence of light

In Fire Bird V, a slotted encoder disc is mounted on the wheel, placed in between the slot of position encoder as shown in Figure 1.1. When encoder disc rotates it cuts IR light alternately because of which photo transistor gets turned on and off and we get logic 0 and 1 as the output alternatively. The output generated will be a square wave signal also known as pulse train. Refer to Figure 3.6



**Figure 3.5: Slotted Encoder Disc**



**Figure 3.6: Square Wave**

**Problem Statement:** Program the robot to move forward, a distance of 50cm.

To make robot move forward for particular distance, first we should know

* the distance to be traversed by robot in each pulse and
* Required pulse count i.e. the number of pulses required for robot to move a specified distance.

**To calculate the distance traversed in each pulse, we should know the following parameters:**

Diameter of the wheel: 5.1cm

Circumference of the wheel: 5.1cm \* 3.14 = 16.014cm = 160.14mm

Number slots on the encoder disc: 30

The distance traversed by robot will be: Circumference of wheel / Number of slots on the

encoder disc

* 160.14 mm / 30 = 5.338mm / pulse.

**Therefore,**

**Required pulse count = Distance to be moved / Distance traversed by robot in each pulse.**

**Hence, for 50cm distance, the required pulse count = 50\*10^1 / 5.338**

**Algorithm:**

1. Start
2. Calculate the required pulse count to move robot 50cm as shown above.
3. Move robot forward
4. Check the current pulse count

i. If it is equal to the required pulse count go to step 5.

ii. If not then go to step 4.

1. Stop.

**Extension:**

Program the robot to take turn of 90 degrees.

Program the robot to trace square of side 40cm using position encoder.

**Note: For more details, you can refer to 8051 Hardware manual section 3.7**.

**Evaluation/Conclusion:**