|  |  |
| --- | --- |
| **SLIDES** | **NARRATION** |
| Introduction to motion control | Hello friends , now we will study about the different motions of Firebird V robot. As there are two DC motors in our robot, we have to learn to perform various motion on motors and that too autonomously.(**press next**) |
| Agenda for discussion | So the agenda for discussion in this module is :   1. Basic movement of robot  * Motion of robot * Understanding L293D IC  1. Motion interfacing on Firebird  * Pin connection * Logic table * Writing C-code(**press next**) |
| Various motions of FIREBIRD V robot | As you can see the various motions of FIREBIRD V robot are(**press next**)   1. FORWARD(**press next**) 2. REVERSE (**press next**) 3. RIGHT TURN (**press next**) 4. (In this case the left wheel moves in forward direction and right wheel moves in reverse direction, due to which robots remains in its position while taking the turn, so it is known as right differential turn) (**press next**) 5. LEFT TURN(**press next**) 6. (In this case the right wheel moves in forward direction and left wheel moves in reverse direction, due to which robots remains in its position while taking the turn, so it is known as left differential turn) (**press next**) (**press next**) 7. Soft right 1(**press next**)   (only the left wheel moves forward) (**press next**)   1. Soft left1(**press next**)   (only the right wheel moves forward) (**press next**)   1. Soft right 2(**press next**)   (only the right wheel moves backward) (**press next**)   1. Soft left 2(**press next**)   (only the left wheel moves backward) (**press next**)  All this turns are self explanatory, just understand them carefully(**press next**) |
| Motors direction | (**press next**)  As there are two terminals of a motor viz. positive and a negative.  So if we connect the positive terminal of the battery to the upper terminal of motor and the negative terminal of the battery to the lower terminal then it moves in the anticlockwise direction(**press next**)  similarly if we change the order of battery terminal then motor will move in clockwise direction. So a single motor will work on 2 bit  1.LOGIC 0 (negative)  2.LOGIC 1 (positive)  Based on the combination of 0 and 1 , motor will move in different direction.(**press next**) |
| Introduction to L293D | When we make a manual robot, we need a remote with a joystick to control the robot. But over here our robots is autonomous, there should not be any manual interference .So who does this work?? The answer is L293D motor driving IC.(**press next**)  This IC takes various inputs in terms of 1 and 0 and control the direction of motor. As it drives the motors, it is known as motor driving IC. This IC can supply around 600 mA of current to each motor.  So L293D is a 16 pin IC. Now let’s have a look towards the function of each pin.   1. Following are the Enable pins that always needs to be set to logic 1.  * PIN1 * PIN9  1. Following are the input pins of L293D IC  * PIN2 * PIN7 * PIN10 * PIN15  1. Following are the Output pins of L293D IC  * PIN3 * PIN6 * PIN11 * PIN14  1. Following are the Voltage pins of L293D IC  * PIN8-To this pin 12v battery supply is connected which is used to drive the motors. * PIN16- This pin take the supply from the system that is used as a supply for the whole IC.  1. Following are the Ground pins of L293D IC  * PIN4 * PIN5 * PIN12 * PIN13   So in this configuration PIN2 and PIN7 are the two input pins for right motor, and PIN3 and PIN6 are the output pins of right motor to which to motor terminals are connected.  So in this configuration PIN10 and PIN15 are the two input pins for left motor, and PIN11 and PIN14 are the output pins of left motor to which to motor terminals are connected.  So this was all about your L293D motor driving IC.(**press next**) |
| Direction control | (**press next**) Three out of four pins for Direction control is connected at PORT0  and one pin is connected at PORT1 of LPC 2148.(**press next**)  P0.22 and P1.21 is used for left motor and P0.10 and P0.11 is used for right motor control(**press next**)  Two Pins for Enabling Motor Driver IC is connected at PORT0(**press next**)  P0.7 is used for left motor enable and P0.21 is used for right motor control(**press next**) |
| Logic table for direction control | Now we will see the logic table for running the motor. It contains the sequence in which command is given to run the motor in particular direction.  (**press next**)  As you can see in the table to move the motor in forward direction, we give logic 0 and logic 1 to P0.22 and P1.21 respectively to move left motor in forward direction.  Similarly logic 1 and logic 0 to P0.10 and P0.11 respectively to move right motor in forward direction. (**press next**)  So just by interchanging the bit value, we can move the motor in different directions. So you can just have a look at the table. (**press next**) (**press next**) (**press next**) (**press next**) (**press next**)  We can stop the motor by giving logic 0 to all the pins. (**press next**)  We pass this commands in terms of hex value, so the hex value for forward will 0x06 and so on. (**press next**) |
| Motion control programming | So after a detail learning of motors, its motion and its driver IC, lets have a look towards a sample code for driving motors. (**press next**)  So perform the following steps to write the code:   1. Include the header files:  * #include<lpc214x.h>(**press next**)      1. Initialize the ports i.e to specify the function of port as GPIO using PINSEL registers 2. Then define whether a port is an input port or an output port. As motor is an output device, we will set :   And we will keep the motors switched OFF initially, so:   * IO0CLR = 0x00400C00; // Initially stop * IO1CLR = 0x00200000;   (**press next**)     1. After initialization ,create different functions for different motions of the motors as follows:  * FORWARD(); * REVERSE(); * LEFT\_TURN(); * RIGHT\_TURN(); * STOP();  1. The most important thing is to introduce a after every function, because if we don’t do that then microcontroller won’t be able to execute two motions immediate one after the another efficiently , so delay is necessary. Define delay in following way:  * \_delay\_ms(1000);   [1000 in brackets indicates the delay of 1000 milli-seconds i.e 1 second, always remember that delay value is passed in milli-seconds]  So this was all about the motion control of Firebird V robot.  Thank you. |
|  |  |