Title of script: Using ADC on Firebird V

Author: Shailesh Jain

Keywords: ADC, register, channels.

|  |  |
| --- | --- |
| Slides | Narration |
| 1. Title | * Hello Friends. * Welcome to the spoken tutorial on Firebird V Robotics Research Platform. * This platform is based on LPC 2148 microcontroller which belongs to ARM architecture based microcontroller family. * In this tutorial, we will study the **Interrupts on Firebird V robot**.   **(press next)** |
| 1. Agenda | * Now let us consider the agenda of discussion in this tutorial. * First Interrupts which includes * What is an interrupt? * And Closed loop programming * Interrupt handling which includes * Sources of interrupt * Position encoder * Interrupt calculation * External interrupt initialization * ISR and * C-code   **(press next)** |
| 3.Introduction of interrupts | * Now we will see what is an interrupt?**(press next)** * Interrupt can be any signal that causes break in continuity of some ongoing process. **(press next)** * **(read slide)**   **(press next)** |
| 4.Closed loop programming | * Here is a brief introduction to closed loop programming**(press next)** * Closed loop systems are those systems in which feedback is present. **(press next)** * This figure shows a closed loop system where A/D converter provides feedback to the cpu. **(press next)** * This feedback is utilized in making decisions. **(press next)** * An open loop system doesn’t have feedback. **(press next)** |
| 5.Sources of Interrupt | * Now here are some sources of interrupt **(press next)** * Lpc2148 has 21 different sources of interrupt**(press next) (press next)** * It has 2 timers and both of them comprise of timer overflow and timer compare interrupt. **(press next)** * There are 2 serial interrupts **(press next)**which can be for wired and wireless communication. **(press next)** * Then there are 4 external hardware interrupts**(press next)** * Lpc2148 has a real time clock which is used for measuring the time. **(press next)** * Few more interrupts are I2C interrupt**(press next)** and WDT(watch dog timer) interrupt. **(press next)** |
| 6.Sources of Interrupt | * **(press next)** * In lpc2148 interrupts are handled by vectored interrupt controller. * This controller classifies the interrupts in 3 categories **(press next)** * first is Fast interrupt request, these interrupts have the highest priority. **(press next)** * next is Vectored interrupt request i.e. vectored IRQ, these interrupts have priority lower than FRQ**(press next)** * now third category is Non vectored IRQ which has the lowest priority. **(press next)** * at a time, VIC can handle 32 interrupt requests. **(press next)** |
| 7.Position encoder | * Now let us learn about position encoders and how interrupts are used in them. **(press next)** * Here is an image showing place of position encoders on Firebird5. **(press next)** * Position encoder consists of optical encoder and slotted disc assembly as shown in figure. **(press next)** * Position Encoder sensors are used to find position of the wheel. **(press next)** * It consists of IR LED and Photodiode mounted facing each other enclosed in plastic body. When the slotted disc rotates it cuts IR illumination from IR LED alternately due to which photo transistor gives square pulse train as output. **(press next)** * When light emitted by the IR LED is blocked because of alternating slots of the encoder disc logic level of the photo diode changes. **(press next)** * This change in the logic level can be sensed by the microcontroller or by discrete hardware because output of the encoder is connected to the interrupt pin of microcontroller. **(press next)** * In lpc2148 based firebird5 robot, left encoder is connected to external interrupt 3 and right encoder is connected to external interrupt 0. **(press next)** |
| 8.some mathematics | * Now let us do some calculation part.This is to calculate the resolution of position encoder**(press next)** * Number of slots in disc are 30**(press next)** * So number of pulses per rotation are also 30**(press next)** * Diameter of wheel is 52 mms**(press next)** * Therefore the resolution will be**(press next)** * Circumference divided by number of slots which is equal to 5.44 mm per pulse.**(press next)** * So total pulse count is equal to distance divided by resolution. **(press next)** |
| 9.Initialization | * To setup interrupts for position encoders few registers are required to initialize. (**press next**) * These registers are- EXTMODE**(press next)** * EXTPOLAR**(press next)** * VICIntSelect**(press next)** * VICVectCntl**(press next)** * VICVectAddr**(press next)** * EXTINT**(press next)** * VICIntEnable * These registers are explained in detail in further slides. **(press next)** |
| 10.EXTMODE | * Here is EXTMODE register which is External interrupt mode register, this register is used to control whether external interrupt is edge or level sensitive.(**press** **next** ) * Here is a table showing bit representation of EXTMODE register. * Only 4 out of 8 bits are of our use i.e. bit 0 to 3 * Bit 0 and Bit 3 is initially set because we want external interrupt 0 and 3 as edge sensitive. **(press next)** * This register is initialized with value 0x09. **(press next)** |
| 11.EXTPOLAR | * Here is EXTPOLAR register which is External interrupt polarity register, this register is used to control which level(high or low) or which edge(rising or falling) on each pin will cause an interrupt.(**press** **next** ) * Here is a table showing bit representation of EXTPOLAR register. * Only 4 out of 8 bits are of our use i.e. bit 0 to 3 * All bits from Bit 0 to Bit 3 is initially reset because we want external interrupts as falling edge sensitive. **(press next)** * This register is initialized with value 0x00. **(press next)** |
| 12.EXTINT | * Now Here is EXTINT register which is External interrupt flag register, this register is used to check which interrupt has occur ascorresponding bit of this register becomes high on occurrence of an interrupt. We can clear the bit by writing 1 to it. **(press** **next** ) * Here is a table showing bit representation of EXTINT register. * Only 4 out of 8 bits are of our use i.e. bit 0 to 3 * Bit 0 and Bit 3 is initially made clear so that occurrence of these interrupts can be detected. **(press next)** * This register is initialized with value 0x09. **(press next)** |
| 13.VICIntSelect | * Here comes VICIntSelect register which is Interrupt select register.   This is a read/write accessible register. It classifies each of the 32 interrupt requests as contributing to FIQ or IRQ. Writing 1 to any bit contributes to FIQ and writing 0 contributes to IRQ. **(press next)**   * Here is a table for bit description of this register, so here you can see that every bit has an associated interrupt with it.   As we want Eint0 and Eint3 as IRQ so we will reset bit 14 and bit 17 of this register. Others can take any value 0 or 1 as per use.**(press next)**   * Here we have initialized all with 0 only. **(press next)** |
| 14.VICIntEnable | * Now we will see VICIntEnable register which is Interrupt enable register, Writing 1 to any bit enables the corresponding interrupt. **(press next)** * Here is a table for bit description of this register which is same as VICIntSelect register.**(press next)** * As we want to enable EINT0 and EINT3 interrupt so the value of this register will be 0x00024000. **(press next)** |
| 15.VICVectCntl | * Let us now see next register, VICVectCntl register which is Vector Control register. There are total 16 vector control registers. Individual register is used for anyone of the vectored IRQ slot. VICVectCntl0 has the highest priority and VICVectCntl15 has the lowest priority. **(press next)** * Here is bit description of this register, bits 6 to 31 are reserved and hence initialized with 0, then bit 5 is used to enable the vectored IRQ slot, its value is kept 1, then bits 0 to 4 define the number of the interrupt request assigned to this vectored IRQ slot.It should be initialized with the interrupt number. **(press next)** * So final value in VICVectCntlx register will be 0x20 or with interrupt number. **(press next)** |
| 16. VICVectAddr | * Now comes the last register, VICVectAddr register which is vector address register. These are also 16 in number and they hold the addresses of the Interrupt Service routines (ISRs) for the 16 vectored IRQ slots. (**press** **next**) * This register is initialized with the address of the IRQ slot. **(press next)** * Now let us see an example, we want to set priority of right encoder i.e. EINT0 higher than the left encoder so in order to achieve this we can use both VICVectCntl and VICVectAddr registers as follows, * As I have mentioned earlier that the interrupt number of EINT3 is 17 and that of EINT0 is 14 so VICVectCntl0 register is initialized with this interrupt number 17 and VICVectCntl1 register with interrupt number 14 in order to have the desired priority. At the same time VICVectAddr registers also need to be initialized with the corresponding IRQ slot addresses. **(press next)** |
| 17.ISR | * Now we will start with interrupt service routine (**press** **next** ) * The format of ISR for external interrupt is as shown**(press next)** * **(press next)**The operation which we want to perform on occurrence of an interrupt must be written inside the ISR at the place where the word code is written. **(press next)** * As we know there are 4 external interrupts so n can take values from 0 to 3. **(press next)** |
| 18. C program | * Now let us see the syntax for C program**(press next)** * First we need to initialize the port pins as external interrupts using PINSEL register. **(press next) (press next)** * Now next step is to initialize the interrupts**(press next)** * Initialize all the required registers as described before.**(press next)** |
| 19.Thank you | * So here we have successfully understood the concept of interrupts and their programming in LPC2148 based Firebird V robot. With this we have come to end to this tutorial. Thank you for listening. For any queries or doubts you can visit <http://qa.e-yantra.org/> * This is Bhumika Varshney Signing off!! |