Interrupts On Firebird-V Robot

e-Yantra Team
Embedded Real-Time Systems Lab
Indian Institute of Technology-Bombay

IIT Bombay
July 2, 2015

Agenda for Discussion

1. Interrupts
   - What is an Interrupt
   - Closed Loop Programming
2. Interrupt-Handling on Firebird
   - Sources of Interrupt
   - Position Encoder
   - Interrupt Calculation
   - External Interrupt Initialization
   - ISR
   - C-Code

What is an Interrupt

What is an Interrupt

✓ Any signal that causes break in continuity of some ongoing process

What is an Interrupt

- ✓ Any signal that causes break in continuity of some ongoing process
- ✓ In microcontrollers interrupt signal halts the execution of main program and dedicates processor to another task

What is an Interrupt

- ✓ Any signal that causes break in continuity of some ongoing process
- ✓ In microcontrollers interrupt signal halts the execution of main program and dedicates processor to another task
- ✓ While main program is running, if an interrupt occurs, execution of main program is stopped, and program counter goes to address of ISR

What is an Interrupt

- ✓ Any signal that causes break in continuity of some ongoing process
- ✓ In microcontrollers interrupt signal halts the execution of main program and dedicates processor to another task
- ✓ While main program is running, if an interrupt occurs, execution of main program is stopped, and program counter goes to address of ISR
- ✓ Interrupt Service Routine: Program that needs to be executed when interrupt occurs

What is an Interrupt

- ✓ Any signal that causes break in continuity of some ongoing process
- ✓ In microcontrollers interrupt signal halts the execution of main program and dedicates processor to another task
- ✓ While main program is running, if an interrupt occurs, execution of main program is stopped, and program counter goes to address of ISR
- ✓ Interrupt Service Routine: Program that needs to be executed when interrupt occurs
- ✓ After program inside ISR is executed completely, program counter returns back to point where main program was interrupted

Closed Loop Programming

Closed Loop Programming

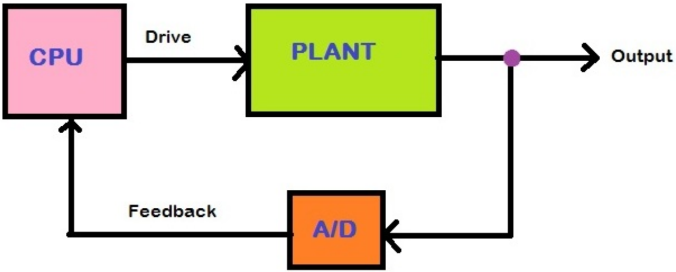
- Systems that utilize feedback are called closed-loop control systems

Closed Loop Programming

- Systems that utilize feedback are called closed-loop control systems

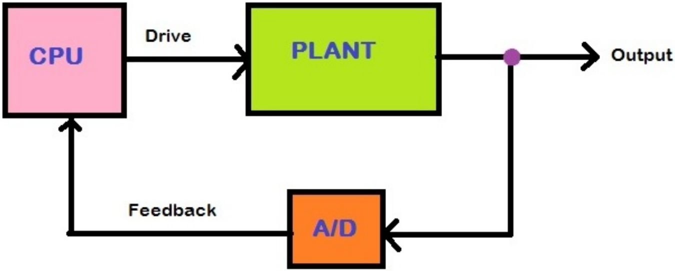
Closed Loop Programming

- Systems that utilize feedback are called closed-loop control systems



Closed Loop Programming

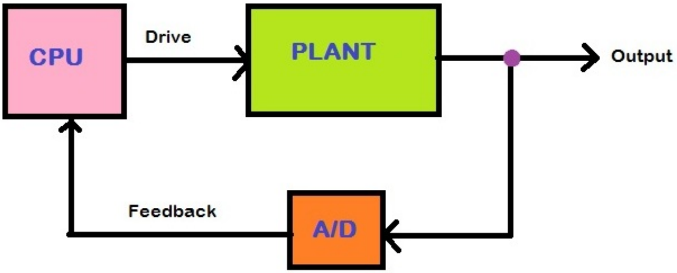
- Systems that utilize feedback are called closed-loop control systems



- The feedback is used to make decisions about changes to the control signal that drives the plant

Closed Loop Programming

- Systems that utilize feedback are called closed-loop control systems



- The feedback is used to make decisions about changes to the control signal that drives the plant
- An open-loop control system doesn't have or doesn't use feedback

Sources of Interrupt on LC2148

Sources of Interrupt on LC2148

LPC2148 has **Twenty-One** Different sources for Interrupt generation

Sources of Interrupt on LC2148

LPC2148 has **Twenty-One** Different sources for Interrupt generation

- Timer Overflow Interrupt

Sources of Interrupt on LC2148

LPC2148 has **Twenty-One** Different sources for Interrupt generation

- Timer Overflow Interrupt
- Timer Compare

Sources of Interrupt on LC2148

LPC2148 has **Twenty-One** Different sources for Interrupt generation

- Timer Overflow Interrupt
- Timer Compare
- Serial interrupt

Sources of Interrupt on LC2148

LPC2148 has **Twenty-One** Different sources for Interrupt generation

- Timer Overflow Interrupt
- Timer Compare
- Serial interrupt
- Wired & Wireless Interrupt

Sources of Interrupt on LC2148

LPC2148 has **Twenty-One** Different sources for Interrupt generation

- Timer Overflow Interrupt
- Timer Compare
- Serial interrupt
- Wired & Wireless Interrupt
- External hardware Interrupt

Sources of Interrupt on LC2148

LPC2148 has **Twenty-One** Different sources for Interrupt generation

- Timer Overflow Interrupt
- Timer Compare
- Serial interrupt
- Wired & Wireless Interrupt
- External hardware Interrupt
- RTC Interrupt

Sources of Interrupt on LC2148

LPC2148 has **Twenty-One** Different sources for Interrupt generation

- Timer Overflow Interrupt
- Timer Compare
- Serial interrupt
- Wired & Wireless Interrupt
- External hardware Interrupt
- RTC Interrupt
- I2C Interrupt

Sources of Interrupt on LC2148

LPC2148 has **Twenty-One** Different sources for Interrupt generation

- Timer Overflow Interrupt
- Timer Compare
- Serial interrupt
- Wired & Wireless Interrupt
- External hardware Interrupt
- RTC Interrupt
- I2C Interrupt
- WDT Interrupt

Sources of Interrupt on LPC2148

Sources of Interrupt on LPC2148

Interrupts in LPC2148 are handled by **Vectored Interrupt Controller(VIC)** and are classified into 3 types based on the priority levels.(Though Interrupts can classified in many different ways as well)

Sources of Interrupt on LPC2148

Interrupts in LPC2148 are handled by **Vectored Interrupt Controller(VIC)** and are classified into 3 types based on the priority levels.(Though Interrupts can classified in many different ways as well)

1. **Fast Interrupt Request i.e FIQ:** which has highest priority

Sources of Interrupt on LPC2148

Interrupts in LPC2148 are handled by **Vectored Interrupt Controller(VIC)** and are classified into 3 types based on the priority levels.(Though Interrupts can classified in many different ways as well)

1. **Fast Interrupt Request i.e FIQ:** which has highest priority
2. **Vectored Interrupt Request i.e Vectored IRQ :** which has 'middle' or priority between FIQ and Non-Vectored IRQ.

Sources of Interrupt on LPC2148

Interrupts in LPC2148 are handled by **Vectored Interrupt Controller(VIC)** and are classified into 3 types based on the priority levels.(Though Interrupts can classified in many different ways as well)

1. **Fast Interrupt Request i.e FIQ:** which has highest priority
2. **Vectored Interrupt Request i.e Vectored IRQ :** which has 'middle' or priority between FIQ and Non-Vectored IRQ.
3. **Non-Vectored IRQ :** which has the lowest priority.

Sources of Interrupt on LPC2148

Interrupts in LPC2148 are handled by **Vectored Interrupt Controller(VIC)** and are classified into 3 types based on the priority levels.(Though Interrupts can classified in many different ways as well)

1. **Fast Interrupt Request i.e FIQ:** which has highest priority
2. **Vectored Interrupt Request i.e Vectored IRQ :** which has 'middle' or priority between FIQ and Non-Vectored IRQ.
3. **Non-Vectored IRQ :** which has the lowest priority.

The Vectored Interrupt Controller (VIC) takes 32 interrupt request inputs.

Sources of Interrupt on LPC2148

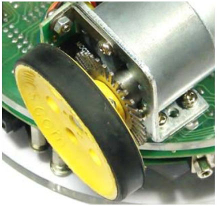
Interrupts in LPC2148 are handled by **Vectored Interrupt Controller(VIC)** and are classified into 3 types based on the priority levels.(Though Interrupts can classified in many different ways as well)

1. **Fast Interrupt Request i.e FIQ:** which has highest priority
2. **Vectored Interrupt Request i.e Vectored IRQ :** which has 'middle' or priority between FIQ and Non-Vectored IRQ.
3. **Non-Vectored IRQ :** which has the lowest priority.

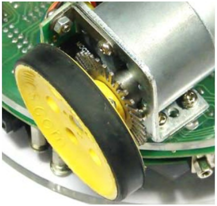
The Vectored Interrupt Controller (VIC) takes 32 interrupt request inputs.

Position encoder

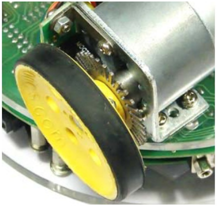
Position encoder



Position encoder

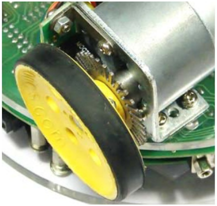


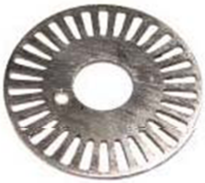
Position encoder



1. Optical position encoders are used for position feedback

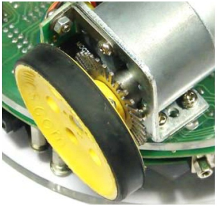
Position encoder





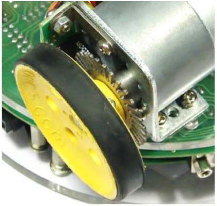
1. Optical position encoders are used for position feedback
2. It consists of IR LED and photo diode placed opposite of each other

Position encoder



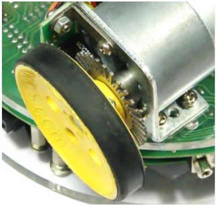
1. Optical position encoders are used for position feedback
2. It consists of IR LED and photo diode placed opposite of each other
3. When IR light is interrupted by encoder disc,its output state changes (high to low or low to high)

Position encoder



1. Optical position encoders are used for position feedback
2. It consists of IR LED and photo diode placed opposite of each other
3. When IR light is interrupted by encoder disc,its output state changes (high to low or low to high)
4. Output of the encoder is connected to the interrupt pin of the microcontroller

Position encoder



1. Optical position encoders are used for position feedback
2. It consists of IR LED and photo diode placed opposite of each other
3. When IR light is interrupted by encoder disc,its output state changes (high to low or low to high)
4. Output of the encoder is connected to the interrupt pin of the microcontroller
5. Left encoder is connected to EINT3 and Right encoder is connected to EINT0

Some Mathematics...

Some Mathematics...

1. Number of slots in disc = 30

Some Mathematics...

1. Number of slots in disc $= 30$
2. Number of Pulse/rotation $= 30$

Some Mathematics...

1. Number of slots in disc = 30
2. Number of Pulse/rotation = 30
3. Diameter of wheel = 52mm

Some Mathematics...

1. Number of slots in disc $= 30$
2. Number of Pulse/rotation $= 30$
3. Diameter of wheel $=$ 52mm
4. Resolution of position encoder

Some Mathematics...

1. Number of slots in disc = 30
2. Number of Pulse/rotation = 30
3. Diameter of wheel = 52mm
4. Resolution of position encoder

Some Mathematics...

1. Number of slots in disc = 30
2. Number of Pulse/rotation = 30
3. Diameter of wheel = 52mm
4. Resolution of position encoder

   $= (\pi * \text{d})/30 = 5.44$

Some Mathematics...

1. Number of slots in disc $= 30$
2. Number of Pulse/rotation $= 30$
3. Diameter of wheel $= 52\text{mm}$
4. Resolution of position encoder

   $$= (\pi * \text{d})/30 = 5.44$$
5. Pulse count

   $$= \text{distance}/5.44$$

Initialization

To setup Interrupts for right and left position encoders, few registers are required to initialize:

Initialization

To setup Interrupts for right and left position encoders, few registers are required to initialize:

- EXTMODE - External Interrupt Mode Register

Initialization

To setup Interrupts for right and left position encoders, few registers are required to initialize:

- EXTMODE - External Interrupt Mode Register
- EXTPOLAR - External Interrupt Polarity Register

Initialization

To setup Interrupts for right and left position encoders, few registers are required to initialize:

- EXTMODE - External Interrupt Mode Register
- EXTPOLAR - External Interrupt Polarity Register
- VICIntSelect - Interrupt Select Register

Initialization

To setup Interrupts for right and left position encoders, few registers are required to initialize:

- EXTMODE - External Interrupt Mode Register
- EXTPOLAR - External Interrupt Polarity Register
- VICIntSelect - Interrupt Select Register
- VICVectCntl - Vector Control Register

Initialization

To setup Interrupts for right and left position encoders, few registers are required to initialize:

- EXTMODE - External Interrupt Mode Register
- EXTPOLAR - External Interrupt Polarity Register
- VICIntSelect - Interrupt Select Register
- VICVectCntl - Vector Control Register
- VICVectAddr - Vector Address Register

Initialization

To setup Interrupts for right and left position encoders, few registers are required to initialize:

- EXTMODE - External Interrupt Mode Register
- EXTPOLAR - External Interrupt Polarity Register
- VICIntSelect - Interrupt Select Register
- VICVectCntl - Vector Control Register
- VICVectAddr - Vector Address Register
- EXTINT - External Interrupt flag Register

Initialization

To setup Interrupts for right and left position encoders, few registers are required to initialize:

- EXTMODE - External Interrupt Mode Register
- EXTPOLAR - External Interrupt Polarity Register
- VICIntSelect - Interrupt Select Register
- VICVectCntl - Vector Control Register
- VICVectAddr - Vector Address Register
- EXTINT - External Interrupt flag Register
- VICIntEnable - Interrupt Enable Register

Initialization

To setup Interrupts for right and left position encoders, few registers are required to initialize:

- EXTMODE - External Interrupt Mode Register
- EXTPOLAR - External Interrupt Polarity Register
- VICIntSelect - Interrupt Select Register
- VICVectCntl - Vector Control Register
- VICVectAddr - Vector Address Register
- EXTINT - External Interrupt flag Register
- VICIntEnable - Interrupt Enable Register

EXTMODE- External Interrupt Mode Register

- This register is Used to control whether external interrupt is edge or level sensitive

EXTMODE- External Interrupt Mode Register

- This register is Used to control whether external interrupt is edge or level sensitive

Bit	Symbol	Description	Bit Value
7	-	Reserved	0
6	-	Reserved	0
5	-	Reserved	0
4	-	Reserved	0
3	EXTMODE3	To select EINT3 as edge or level sensitive	1
2	EXTMODE2	To select EINT2 as edge or level sensitive	0
1	EXTMODE1	To select EINT1 as edge or level sensitive	0
0	EXTMODE0	To select EINT0 as edge or level sensitive	1

EXTMODE- External Interrupt Mode Register

- This register is Used to control whether external interrupt is edge or level sensitive

Bit	Symbol	Description	Bit Value
7	-	Reserved	0
6	-	Reserved	0
5	-	Reserved	0
4	-	Reserved	0
3	EXTMODE3	To select EINT3 as edge or level sensitive	1
2	EXTMODE2	To select EINT2 as edge or level sensitive	0
1	EXTMODE1	To select EINT1 as edge or level sensitive	0
0	EXTMODE0	To select EINT0 as edge or level sensitive	1

EICRA = 0x00

EXTPOLAR- External Interrupt Polarity Register

- This register is Used to control which level(high or low) or edge(rising or falling) on each pin will cause an interrupt

EXTPOLAR- External Interrupt Polarity Register

- This register is Used to control which level(high or low) or edge(rising or falling) on each pin will cause an interrupt

Bit	Symbol	Description	Bit Value
7	-	Reserved	0
6	-	Reserved	0
5	-	Reserved	0
4	-	Reserved	0
3	EXTPOLAR3	EINT3 as falling edge sensitive	0
2	EXTPOLAR2	EINT2 as falling edge sensitive	0
1	EXTPOLAR1	EINT1 as falling edge sensitive	0
0	EXTPOLAR0	EINT0 as falling edge sensitive	0

EXTPOLAR- External Interrupt Polarity Register

- This register is Used to control which level(high or low) or edge(rising or falling) on each pin will cause an interrupt

Bit	Symbol	Description	Bit Value
7	-	Reserved	0
6	-	Reserved	0
5	-	Reserved	0
4	-	Reserved	0
3	EXTPOLAR3	EINT3 as falling edge sensitive	0
2	EXTPOLAR2	EINT2 as falling edge sensitive	0
1	EXTPOLAR1	EINT1 as falling edge sensitive	0
0	EXTPOLAR0	EINT0 as falling edge sensitive	0

EXTPOLAR=0x00 ;

EXTINT- External Interrupt Flag Register

- When Interrupt occurs, corresponding bit of this register becomes high. To clear it, write 1 to that bit.

EXTINT- External Interrupt Flag Register

- When Interrupt occurs, corresponding bit of this register becomes high. To clear it, write 1 to that bit.

Bit	Symbol	Description	Bit Value
7	-	Reserved	0
6	-	Reserved	0
5	-	Reserved	0
4	-	Reserved	0
3	EINT3	indicate EINT3 interrupt occurence	1
2	EINT2	indicate EINT2 interrupt occurence	0
1	EINT1	indicate EINT1 interrupt occurence	0
0	EINT0	indicate EINT0 interrupt occurence	1

EXTINT- External Interrupt Flag Register

- When Interrupt occurs, corresponding bit of this register becomes high. To clear it, write 1 to that bit.

Bit	Symbol	Description	Bit Value
7	-	Reserved	0
6	-	Reserved	0
5	-	Reserved	0
4	-	Reserved	0
3	EINT3	indicate EINT3 interrupt occurence	1
2	EINT2	indicate EINT2 interrupt occurence	0
1	EINT1	indicate EINT1 interrupt occurence	0
0	EINT0	indicate EINT0 interrupt occurence	1

EXTINT = 0x09 ;

VICIntSelect- Interrupt Select Register

- This register 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.

VICIntSelect- Interrupt Select Register

- This register 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.

Bit	31-23	22	21	20	19	18	17	16	15
Symbol	-	USB	AD1	BOD	I2C1	ED0	EINT3	EINT2	EINT1

Bit	14	13	12	11	10	9	8	7
Symbol	EINT0	RTC	PLL	SPI1/SSP	SPI0	I2C0	PWM0	UART1

Bit	6	5	4	3	2	1	0
Symbol	UART0	TIMER1	TIMER0	ARMCore1	ARMCore0	-	WDT

VICIntSelect=0x00000000 ;

VICIntEnable- Interrupt Enable Register

- This register classifies each of the 32 interrupt requests as contributing to FIQ or IRQ. Writing 1 to any bit enables the corresponding interrupt.

VICIntEnable- Interrupt Enable Register

- This register classifies each of the 32 interrupt requests as contributing to FIQ or IRQ. Writing 1 to any bit enables the corresponding interrupt.

Bit	31-23	22	21	20	19	18	17	16	15
Symbol	-	USB	AD1	BOD	I2C1	ED0	EINT3	EINT2	EINT1

Bit	14	13	12	11	10	9	8	7
Symbol	EINT0	RTC	PLL	SPI1/SSP	SPI0	I2C0	PWM0	UART1

Bit	6	5	4	3	2	1	0
Symbol	UART0	TIMER1	TIMER0	ARMCore1	ARMCore0	-	WDT

VICIntEnable=0x00024000 ;

VICVectCntl- Vector Control Registers0-15

- There are total 16 vector control registers. Each of these registers controls one of the 16 vectored IRQ slots. VICVectCntl0 has the highest priority and VICVectCntl15 the lowest.

VICVectCntl- Vector Control Registers0-15

- There are total 16 vector control registers. Each of these registers controls one of the 16 vectored IRQ slots. VICVectCntl0 has the highest priority and VICVectCntl15 the lowest.

Bit	Symbol	Description	Bit Value
31-6	-	Reserved	0
5	IRQslot_en	When 1, this vectored IRQ slot is enabled	1
4-0	int_request/ sw_int_assig	The number of the interrupt request assigned to this slot	Interrupt number

VICVectCntl- Vector Control Registers0-15

- There are total 16 vector control registers. Each of these registers controls one of the 16 vectored IRQ slots. VICVectCntl0 has the highest priority and VICVectCntl15 the lowest.

Bit	Symbol	Description	Bit Value
31-6	-	Reserved	0
5	IRQslot_en	When 1, this vectored IRQ slot is enabled	1
4-0	int_request/ sw_int_assig	The number of the interrupt request assigned to this slot	Interrupt number

VICVectCntlx=0x20|Interrupt number

VICVectAddr- Vector Address Registers0-15

- These registers hold the addresses of the Interrupt Service routines (ISRs) for the 16 vectored IRQ slots.

VICVectAddr- Vector Address Registers0-15

- These registers hold the addresses of the Interrupt Service routines (ISRs) for the 16 vectored IRQ slots.

  VICVectCntlx=0x20|Interrupt number

For Example:

VICVectAddr- Vector Address Registers0-15

- These registers hold the addresses of the Interrupt Service routines (ISRs) for the 16 vectored IRQ slots.

  VICVectCntlx=0x20|Interrupt number

For Example:

To set EINT3(left encoder) at highest priority and EINT0(right encoder) at lower priority, combination of both VICVectCntl and VICVectAddr registers are used as follows -

```
VICVectCntl0 = 0x20|17;
VICVectAddr0 = (int)IRQ_Eint3;
VICVectCntl1 = 0x20|14;
VICVectAddr1 = (int)IRQ_Eint0;
```

ISR-Interrupt Service Routine

ISR-Interrupt Service Routine

The format of ISR for external interrupt is

ISR-Interrupt Service Routine

The format of ISR for external interrupt is

ISR Format

ISR-Interrupt Service Routine

The format of ISR for external interrupt is

ISR Format

```
__irq IRQ_Eintn
{
    code
}
```

ISR-Interrupt Service Routine

The format of ISR for external interrupt is

ISR Format

```
__irq IRQ_Eintn
{
    code
}
```

Where n = External Interrupt Number (For LPC2148: n=0-4)

Syntax for C-Program

Syntax for C-Program

- Port Initialization

Left and Right Encoder Port Initialization

Syntax for C-Program

- Port Initialization

Left and Right Encoder Port Initialization

```
PINSEL1 = 0x20000001; //Enabling P0.16 as EINT0  and P0.30 as EINT3
```

Syntax for C-Program

- Port Initialization

Left and Right Encoder Port Initialization

```
PINSEL1 = 0x20000001; //Enabling P0.16 as EINT0  and P0.30 as EINT3
```

- Interrupt Initialization

Left and right Encoder Interrupt Initialization

Syntax for C-Program

- Port Initialization

Left and Right Encoder Port Initialization

```
PINSEL1 = 0x20000001; //Enabling P0.16 as EINT0  and P0.30 as EINT3
```

- Interrupt Initialization

Left and right Encoder Interrupt Initialization

```
EXTMODE = 0x09;                // EINT3 and EINT0 is edge sensitive
EXTPOLAR = 0x00;                // EINT3 and EINT0 in triggered on falling edge
VICIntSelect = 0x00000000;      // Setting EINT2 and EINt0 as IRQ(Vectored)
EXTINT = 0x09;                   //Setting EINT2 & EINT0 interrupt flag
VICIntEnable = (1<<17) | (1<<14); // Enable EINT3 & EINT0 flags
```

Syntax for C-Program

- Port Initialization

Left and Right Encoder Port Initialization

```
PINSEL1 = 0x20000001; //Enabling P0.16 as EINT0  and P0.30 as EINT3
```

- Interrupt Initialization

Left and right Encoder Interrupt Initialization

```
EXTMODE = 0x09;                // EINT3 and EINT0 is edge sensitive
EXTPOLAR = 0x00;                // EINT3 and EINT0 in triggered on falling edge
VICIntSelect = 0x00000000;      // Setting EINT2 and EINt0 as IRQ(Vectored)
EXTINT = 0x09;                   //Setting EINT2 & EINT0 interrupt flag
VICIntEnable = (1<<17) | (1<<14); // Enable EINT3 & EINT0 flags
```

Thank You!

Post your queries on: http://qa.e-yantra.org/