Basic IO Interfacing on Firebird-V

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

IIT Bombay
July 3, 2015

Agenda for Discussion

1. Input-Output Ports in LPC2148
   - Overview of Ports
   - Acessing Ports
   - Example
2. Write Your First Embedded C Program
   - Buzzer Interfacing
   - Programming Tools
   - C-code

What are Ports?

What are Ports?

1. Junctions where peripheral devices are connected.

What are Ports?

1. Junctions where peripheral devices are connected.
2. Peripheral devices can be

What are Ports?

1. Junctions where peripheral devices are connected.
2. Peripheral devices can be
    1. Input Device

       Example: Switch, Sensors, etc...

What are Ports?

1. Junctions where peripheral devices are connected.
2. Peripheral devices can be
    1. Input Device

       Example: Switch, Sensors, etc...
    2. Output Device

       Example: Buzzer,LCD,Motors,LED, etc...

PORTS in LPC2148

PORTS in LPC2148

1. LPC stands for low power consumption.

PORTS in LPC2148

1. LPC stands for low power consumption.
2. LPC2148 is 64 pin controller

PORTS in LPC2148

1. LPC stands for low power consumption.
2. LPC2148 is 64 pin controller
3. 44 pins can be used as Input/Output Pins

PORTS in LPC2148

1. LPC stands for low power consumption.
2. LPC2148 is 64 pin controller
3. 44 pins can be used as Input/Output Pins
4. Pins are grouped together and called as PORT

PORTS in LPC2148

1. LPC stands for low power consumption.
2. LPC2148 is 64 pin controller
3. 44 pins can be used as Input/Output Pins
4. Pins are grouped together and called as PORT
5. LPC2148 has Two 32-bit ports

   PORTx; x = 0 or 1

PORTS in LPC2148

1. LPC stands for low power consumption.
2. LPC2148 is 64 pin controller
3. 44 pins can be used as Input/Output Pins
4. Pins are grouped together and called as PORT
5. LPC2148 has Two 32-bit ports

   PORTx; x = 0 or 1

   1. PORT 0- Out of 32 pins, 24th, 26th and 27th pins are not available and 31st pin can only be used as output pin.

PORTS in LPC2148

1. LPC stands for low power consumption.
2. LPC2148 is 64 pin controller
3. 44 pins can be used as Input/Output Pins
4. Pins are grouped together and called as PORT
5. LPC2148 has Two 32-bit ports

   PORTx; x = 0 or 1

   1. PORT 0- Out of 32 pins, 24th, 26th and 27th pins are not available and 31st pin can only be used as output pin.
   2. PORT 1- Out of 32 pins, 0 to 15 pins are not available.

Accessing PORTS

Accessing PORTS

1. Each Ports has five associated registers with it.

Accessing PORTS

1. Each Ports has five associated registers with it.
    1. PINSELx x = 0,1 and 2

Accessing PORTS

1. Each Ports has five associated registers with it.
    1. PINSELx x = 0,1 and 2
    2. IOxDIR x = 0 or 1

Accessing PORTS

1. Each Ports has five associated registers with it.
    1. PINSELx x = 0,1 and 2
    2. IOxDIR x = 0 or 1
    3. IOxPIN x = 0 or 1

Accessing PORTS

1. Each Ports has five associated registers with it.
    1. PINSELx x = 0,1 and 2
    2. IOxDIR x = 0 or 1
    3. IOxPIN x = 0 or 1
    4. IOxSET x = 0 or 1

Accessing PORTS

1. Each Ports has five associated registers with it.
    1. PINSELx — x = 0,1 and 2
    2. IOxDIR — x = 0 or 1
    3. IOxPIN — x = 0 or 1
    4. IOxSET — x = 0 or 1
    5. IOxCLR — x = 0 or 1

Understanding PINSELx Register

Understanding PINSELx Register

1. Pin function Select Register

Understanding PINSELx Register

1. Pin function Select Register
2. Purpose: To select the function of individual pins

Understanding PINSELx Register

1. Pin function Select Register
2. Purpose: To select the function of individual pins
3. PINSEL0 and PINSEL1 is used for PORT0

Understanding PINSELx Register

1. Pin function Select Register
2. Purpose: To select the function of individual pins
3. PINSEL0 and PINSEL1 is used for PORT0
4. PINSEL2 is used for PORT1

Understanding PINSELx Register

1. Pin function Select Register
2. Purpose: To select the function of individual pins
3. PINSEL0 and PINSEL1 is used for PORT0
4. PINSEL2 is used for PORT1
5. Each pin can be used for atmost 4 functions so in PINSEL register 2 bits are provided for every single pin.

Understanding PINSELx Register

Understanding PINSELx Register

1. PINSEL0 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P0.15		P0.14		...	...	P0.1		P0.0	

Understanding PINSELx Register

1. PINSEL0 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P0.15		P0.14		...	...	P0.1		P0.0	

Understanding PINSELx Register

1. PINSEL0 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P0.15		P0.14		...	...	P0.1		P0.0	

2. PINSEL1 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P0.31		P0.30		...	...	P0.17		P0.16	

Understanding PINSELx Register

1. PINSEL0 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P0.15		P0.14		...	...	P0.1		P0.0	

2. PINSEL1 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P0.31		P0.30		...	...	P0.17		P0.16	

Understanding PINSELx Register

1. PINSEL0 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P0.15		P0.14		...	...	P0.1		P0.0	

2. PINSEL1 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P0.31		P0.30		...	...	P0.17		P0.16	

3. PINSEL2 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P1.31		P1.30		...	...	P1.17		P1.16	

Understanding PINSELx Register

1. PINSEL0 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P0.15		P0.14		...	...	P0.1		P0.0	

2. PINSEL1 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P0.31		P0.30		...	...	P0.17		P0.16	

3. PINSEL2 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P1.31		P1.30		...	...	P1.17		P1.16	

Understanding PINSELx Register

1. PINSEL0 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P0.15		P0.14		...	...	P0.1		P0.0	

2. PINSEL1 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P0.31		P0.30		...	...	P0.17		P0.16	

3. PINSEL2 REGISTER

D31	D30	D29	D28	...	...	D3	D2	D1	D0
P1.31		P1.30		...	...	P1.17		P1.16	

Understanding PINSELx Register

Understanding PINSELx Register

- For Example-

  Consider P0.0 pin- to select function for P0.0 we require D0 and D1 of PINSEL0 register.

Understanding PINSELx Register

- For Example-

  Consider P0.0 pin- to select function for P0.0 we require D0 and D1 of PINSEL0 register.

Understanding PINSELx Register

- For Example-

  Consider P0.0 pin- to select function for P0.0 we require D0 and D1 of PINSEL0 register.

D1 D0	Function
00	GPIO
01	UART0(TxD)
10	PWM1
11	Reserved

Understanding PINSELx Register

- For Example-

  Consider P0.0 pin- to select function for P0.0 we require D0 and D1 of PINSEL0 register.

D1 D0	Function
00	GPIO
01	UART0(TxD)
10	PWM1
11	Reserved

- To set all pins of PORT 0 as GPIO,
  PINSEL0= 0x00000000; To set P0.0 to P0.15 pins as GPIO
  PINSEL1= 0x00000000; To set P0.16 to P0.31 pins as GPIO

Understanding IOxDIR Register

Understanding IOxDIR Register

1. GPIO port direction control register

Understanding IOxDIR Register

1. GPIO port direction control register
2. Purpose: To control the direction of each port pin

Understanding IOxDIR Register

1. GPIO port direction control register
2. Purpose: To control the direction of each port pin
   - IOxDIR=0 ; PORTx is defined as INPUT

Understanding IOxDIR Register

1. GPIO port direction control register
2. Purpose: To control the direction of each port pin
   - IOxDIR=0 ; PORTx is defined as INPUT
   - IOxDIR=1 ; PORTx is defined as OUTPUT

Understanding IOxDIR Register

1. GPIO port direction control register
2. Purpose: To control the direction of each port pin
   - IOxDIR=0 ; PORTx is defined as INPUT
   - IOxDIR=1 ; PORTx is defined as OUTPUT
3. Example:

   For Port0 make lower word as input and upper word as output

Understanding IOxDIR Register

1. GPIO port direction control register
2. Purpose: To control the direction of each port pin
   - IOxDIR=0 ; PORTx is defined as INPUT
   - IOxDIR=1 ; PORTx is defined as OUTPUT
3. Example:

   For Port0 make lower word as input and upper word as output

Understanding IOxDIR Register

1. GPIO port direction control register
2. Purpose: To control the direction of each port pin
   - IOxDIR=0 ; PORTx is defined as INPUT
   - IOxDIR=1 ; PORTx is defined as OUTPUT
3. Example:

   For Port0 make lower word as input and upper word as output

```
PINSEL0=0x00000000;
PINSEL1=0x00000000;
IO0DIR=0xFFFF0000;
```

Understanding IOxPIN Register

Understanding IOxPIN Register

1. GPIO port pin value register

Understanding IOxPIN Register

1. GPIO port pin value register
2. Purpose: To read the current status of port and to write data on port regardless of the pin direction

Understanding IOxPIN Register

1. GPIO port pin value register
2. Purpose: To read the current status of port and to write data on port regardless of the pin direction
3. Save value of a register in a variable

Understanding IOxPIN Register

1. GPIO port pin value register
2. Purpose: To read the current status of port and to write data on port regardless of the pin direction
3. Save value of a register in a variable
4. Example: Read data from Port0

Understanding IOxPIN Register

1. GPIO port pin value register
2. Purpose: To read the current status of port and to write data on port regardless of the pin direction
3. Save value of a register in a variable
4. Example: Read data from Port0

Understanding IOxPIN Register

1. GPIO port pin value register
2. Purpose: To read the current status of port and to write data on port regardless of the pin direction
3. Save value of a register in a variable
4. Example: Read data from Port0

   Suppose value of PORT0 is 0xFFFF0000

Understanding IOxPIN Register

1. GPIO port pin value register
2. Purpose: To read the current status of port and to write data on port regardless of the pin direction
3. Save value of a register in a variable
4. Example: Read data from Port0

   Suppose value of PORT0 is 0xFFFF0000
   x=IO0PIN;

Understanding IOxPIN Register

1. GPIO port pin value register
2. Purpose: To read the current status of port and to write data on port regardless of the pin direction
3. Save value of a register in a variable
4. Example: Read data from Port0

   Suppose value of PORT0 is 0xFFFF0000
   x=IO0PIN;
   then x=0xFFFF0000

Understanding IOxPIN Register

1. GPIO port pin value register
2. Purpose: To read the current status of port and to write data on port regardless of the pin direction
3. Save value of a register in a variable
4. Example: Read data from Port0

   Suppose value of PORT0 is 0xFFFF0000
   x=IO0PIN;
   then x=0xFFFF0000

Understanding IOxSET Register

Understanding IOxSET Register

1. GPIO port output set register

Understanding IOxSET Register

1. GPIO port output set register
2. Purpose: To set logic 1 on desired pins

Understanding IOxSET Register

1. GPIO port output set register
2. Purpose: To set logic 1 on desired pins
3. Writing ones produces highs at the corresponding port pins, writing zeroes has no effect.

Understanding IOxSET Register

1. GPIO port output set register
2. Purpose: To set logic 1 on desired pins
3. Writing ones produces highs at the corresponding port pins, writing zeroes has no effect.
4. Example: To set bit 0 of PORT0(P0.0),

Understanding IOxSET Register

1. GPIO port output set register
2. Purpose: To set logic 1 on desired pins
3. Writing ones produces highs at the corresponding port pins, writing zeroes has no effect.
4. Example: To set bit 0 of PORT0(P0.0),

Understanding IOxSET Register

1. GPIO port output set register
2. Purpose: To set logic 1 on desired pins
3. Writing ones produces highs at the corresponding port pins, writing zeroes has no effect.
4. Example: To set bit 0 of PORT0(P0.0),

   IO0SET=0x00000001;

Understanding IOxSET Register

1. GPIO port output set register
2. Purpose: To set logic 1 on desired pins
3. Writing ones produces highs at the corresponding port pins, writing zeroes has no effect.
4. Example: To set bit 0 of PORT0(P0.0),

   IO0SET=0x00000001;

   Suppose value of PORT0 was 0xFFFF0000, then after execution of the above instruction, value of PORT0 will be 0xFFFF0001

Understanding IOxSET Register

1. GPIO port output set register
2. Purpose: To set logic 1 on desired pins
3. Writing ones produces highs at the corresponding port pins, writing zeroes has no effect.
4. Example: To set bit 0 of PORT0(P0.0),

   IO0SET=0x00000001;

   Suppose value of PORT0 was 0xFFFF0000, then after execution of the above instruction, value of PORT0 will be 0xFFFF0001

Understanding IOxCLR Register

Understanding IOxCLR Register

1. GPIO port output Clear register

Understanding IOxCLR Register

1. GPIO port output Clear register
2. Purpose: To set logic 0 on desired pins

Understanding IOxCLR Register

1. GPIO port output Clear register
2. Purpose: To set logic 0 on desired pins
3. Writing ones produces lows at the corresponding port pins, writing zeroes has no effect.

Understanding IOxCLR Register

1. GPIO port output Clear register
2. Purpose: To set logic 0 on desired pins
3. Writing ones produces lows at the corresponding port pins, writing zeroes has no effect.
4. Example: To reset bit 31 of PORT0(P0.31),

Understanding IOxCLR Register

1. GPIO port output Clear register
2. Purpose: To set logic 0 on desired pins
3. Writing ones produces lows at the corresponding port pins, writing zeroes has no effect.
4. Example: To reset bit 31 of PORT0(P0.31),

Understanding IOxCLR Register

1. GPIO port output Clear register
2. Purpose: To set logic 0 on desired pins
3. Writing ones produces lows at the corresponding port pins, writing zeroes has no effect.
4. Example: To reset bit 31 of PORT0(P0.31),

   IO0CLR=0x80000000;

Understanding IOxCLR Register

1. GPIO port output Clear register
2. Purpose: To set logic 0 on desired pins
3. Writing ones produces lows at the corresponding port pins, writing zeroes has no effect.
4. Example: To reset bit 31 of PORT0(P0.31),

   IO0CLR=0x80000000;

   Suppose value of PORT0 was 0xFFFF0000, then after execution of the above instruction, value of PORT0 will be 0x7FFF0000

Understanding IOxCLR Register

1. GPIO port output Clear register
2. Purpose: To set logic 0 on desired pins
3. Writing ones produces lows at the corresponding port pins, writing zeroes has no effect.
4. Example: To reset bit 31 of PORT0(P0.31),

   IO0CLR=0x80000000;

   Suppose value of PORT0 was 0xFFFF0000, then after execution of the above instruction, value of PORT0 will be 0x7FFF0000

Example

1. Example: Make PORT0 as output port and send bit pattern of 0000FFFF

Example

1. Example: Make PORT0 as output port and send bit pattern of 0000FFFF

Example

1. Example: Make PORT0 as output port and send bit pattern of 0000FFFF
   - Step 1: Select Port 0 as GPIO port

     PINSEL0=0x00000000;
     PINSEL1=0x00000000;

Example

1. Example: Make PORT0 as output port and send bit pattern of 0000FFFF
   - Step 1: Select Port 0 as GPIO port

     PINSEL0=0x00000000;
     PINSEL1=0x00000000;

Example

1. Example: Make PORT0 as output port and send bit pattern of 0000FFFF
   - Step 1: Select Port 0 as GPIO port

     PINSEL0=0x00000000;
     PINSEL1=0x00000000;
   - Step 2: Make PORT0 as output port

     IO0DIR=0xFFFFFFFF;

Example

1. Example: Make PORT0 as output port and send bit pattern of 0000FFFF

   - Step 1: Select Port 0 as GPIO port

     PINSEL0=0x00000000;
     PINSEL1=0x00000000;

   - Step 2: Make PORT0 as output port

     IO0DIR=0xFFFFFFFF;

Example

1. Example: Make PORT0 as output port and send bit pattern of 0000FFFF

   - Step 1: Select Port 0 as GPIO port

     ```
     PINSEL0=0x00000000;
     PINSEL1=0x00000000;
     ```

   - Step 2: Make PORT0 as output port

     ```
     IO0DIR=0xFFFFFFFF;
     ```

   - Step 3: Put data on the Port 0

     ```
     IO0SET=0x0000FFFF;
     IO0CLR=0xFFFF0000;
     ```

Example

1. Example: Make PORT0 as output port and send bit pattern of 0000FFFF

   - Step 1: Select Port 0 as GPIO port

     PINSEL0=0x00000000;
     PINSEL1=0x00000000;

   - Step 2: Make PORT0 as output port

     IO0DIR=0xFFFFFFFF;

   - Step 3: Put data on the Port 0

     IO0SET=0x0000FFFF;
     IO0CLR=0xFFFF0000;

Example

1. Example: Make PORT0 as output port and send bit pattern of 0000FFFF

   - Step 1: Select Port 0 as GPIO port

     PINSEL0=0x00000000;
     PINSEL1=0x00000000;

   - Step 2: Make PORT0 as output port

     IO0DIR=0xFFFFFFFF;

   - Step 3: Put data on the Port 0

     IO0SET=0x0000FFFF;
     IO0CLR=0xFFFF0000;

     OR

     IO0PIN=0x0000FFFF;

Example

1. Example: Make PORT0 as output port and send bit pattern of 0000FFFF

   - Step 1: Select Port 0 as GPIO port

     PINSEL0=0x00000000;
     PINSEL1=0x00000000;

   - Step 2: Make PORT0 as output port

     IO0DIR=0xFFFFFFFF;

   - Step 3: Put data on the Port 0

     IO0SET=0x0000FFFF;
     IO0CLR=0xFFFF0000;

     OR

     IO0PIN=0x0000FFFF;

Buzzer Interfacing in Firebird V

Buzzer Interfacing in Firebird V

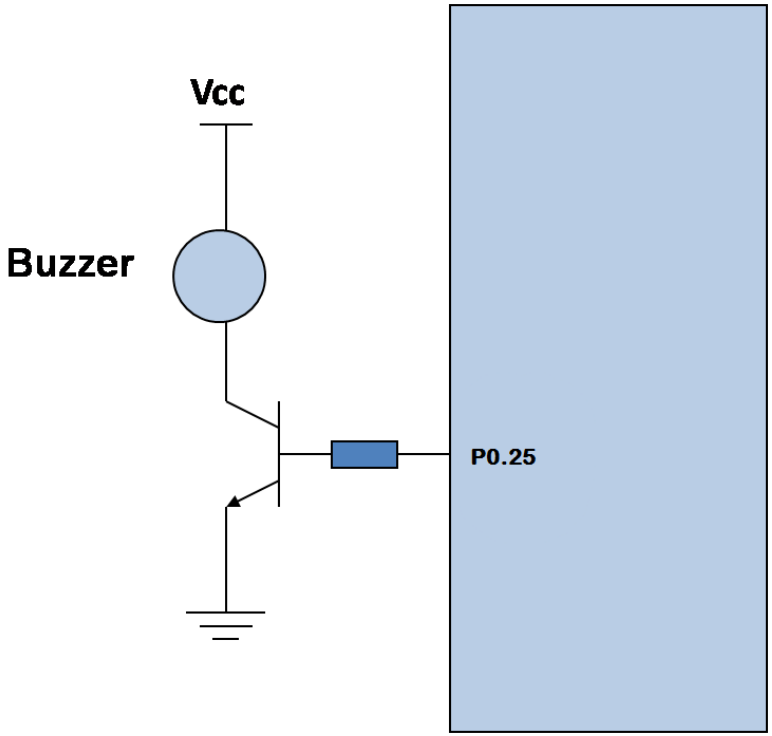
1. Buzzer Connected to Port0 pin 25

Buzzer Interfacing in Firebird V

1. Buzzer Connected to Port0 pin 25

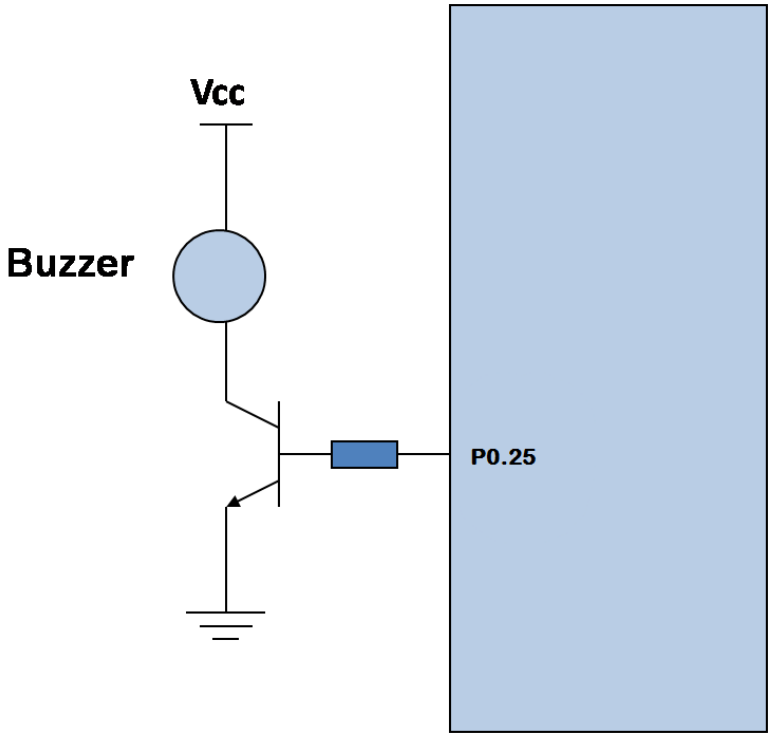
Buzzer Interfacing in Firebird V

1. Buzzer Connected to Port0 pin 25



Buzzer Interfacing in Firebird V

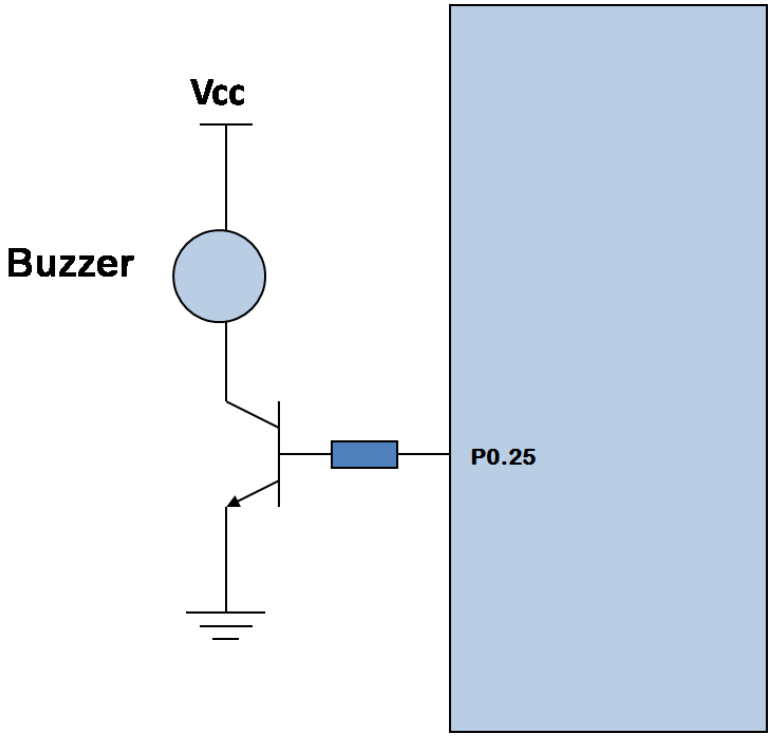
1. Buzzer Connected to Port0 pin 25



2. To Turn on buzzer:

Buzzer Interfacing in Firebird V

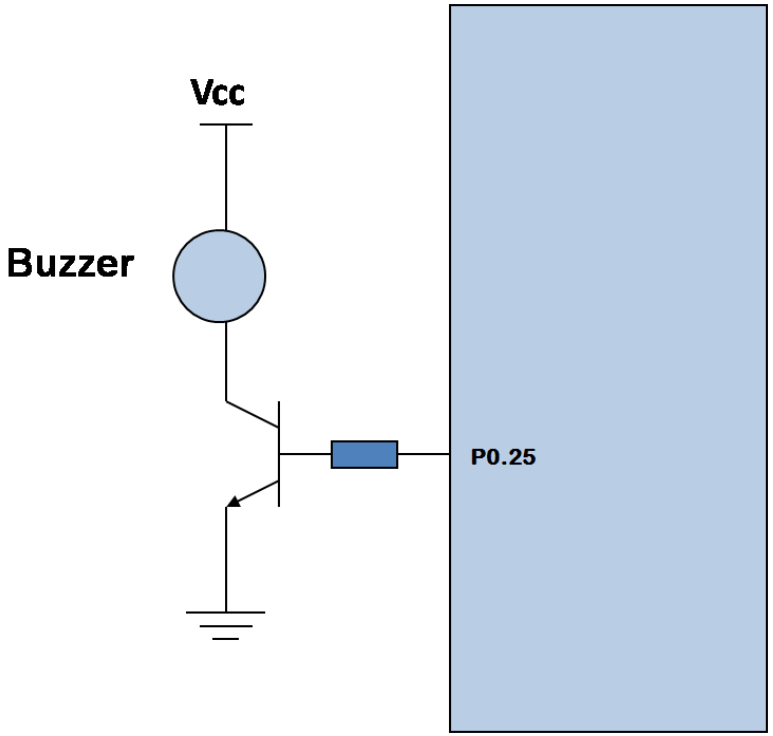
1. Buzzer Connected to Port0 pin 25



2. To Turn on buzzer:

Buzzer Interfacing in Firebird V

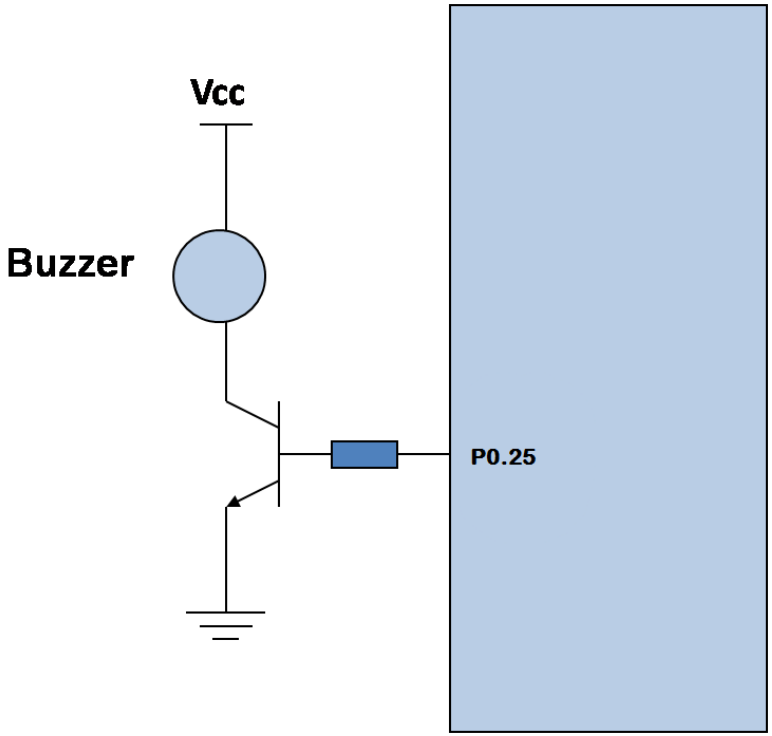
1. Buzzer Connected to Port0 pin 25



2. To Turn on buzzer: send logic HIGH on pin25 of Port0

Buzzer Interfacing in Firebird V

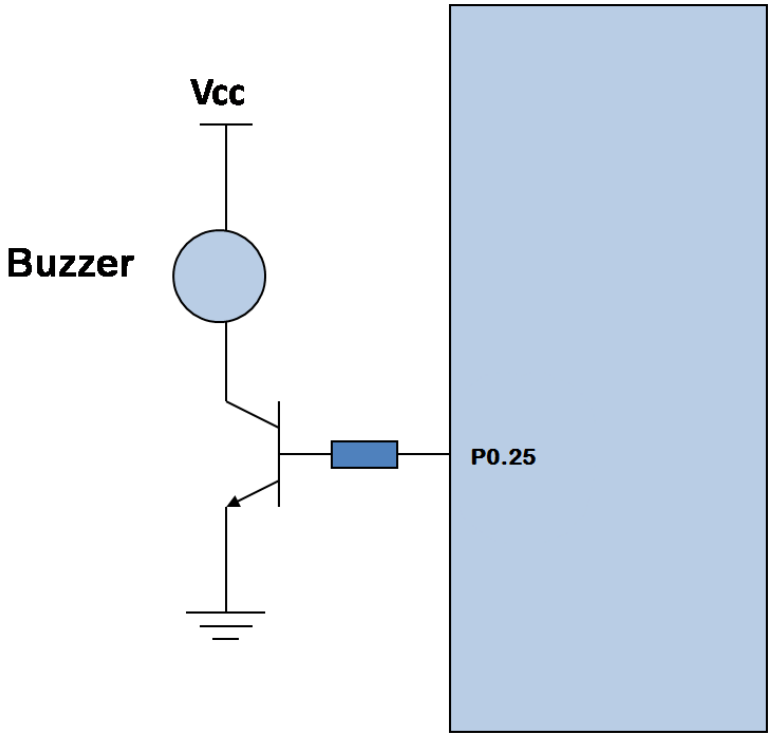
1. Buzzer Connected to Port0 pin 25



2. To Turn on buzzer: send logic HIGH on pin25 of Port0
3. To Turn off buzzer:

Buzzer Interfacing in Firebird V

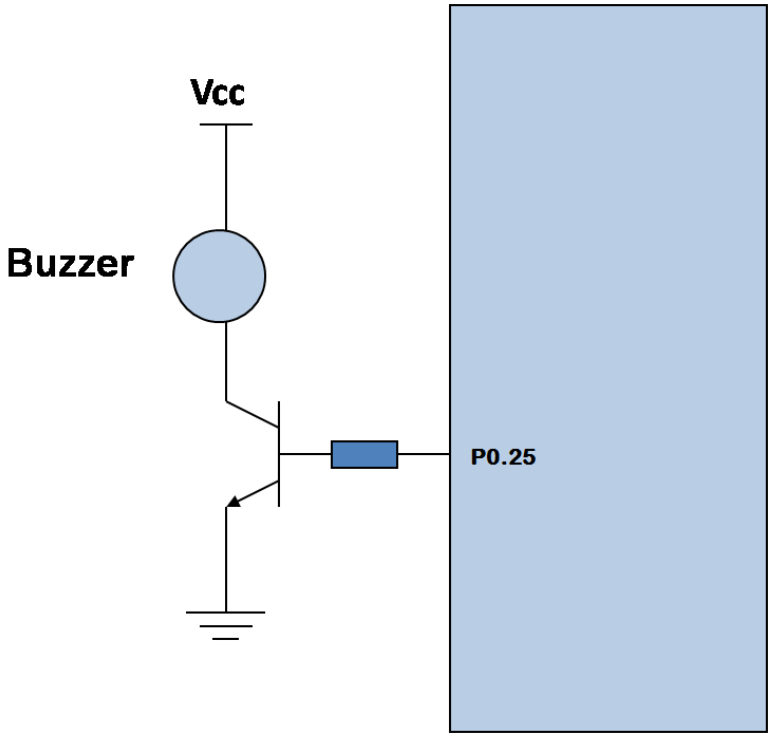
1. Buzzer Connected to Port0 pin 25



2. To Turn on buzzer: send logic HIGH on pin25 of Port0
3. To Turn off buzzer:

Buzzer Interfacing in Firebird V

1. Buzzer Connected to Port0 pin 25



2. To Turn on buzzer: send logic HIGH on pin25 of Port0
3. To Turn off buzzer: send logic LOW on pin25 of Port0

Buzzer Program

Buzzer Program

1. Set P0.25 as GPIO

Buzzer Program

1. Set P0.25 as GPIO

Buzzer Program

1. Set P0.25 as GPIO

```
PINSEL1= 0x00000000;
```

Buzzer Program

1. Set P0.25 as GPIO

   PINSEL1= 0x00000000;

2. Configure PORT0.25 pin as output.

Buzzer Program

1. Set P0.25 as GPIO

   PINSEL1= 0x00000000;

2. Configure PORT0.25 pin as output.

Buzzer Program

1. Set P0.25 as GPIO

   PINSEL1= 0x00000000;

2. Configure PORT0.25 pin as output.

   IO0DIR= 0x02000000;

Buzzer Program

1. Set P0.25 as GPIO

   PINSEL1= 0x00000000;

2. Configure PORT0.25 pin as output.

   IO0DIR= 0x02000000;

3. To turn ON the buzzer set P0.25 output high

Buzzer Program

1. Set P0.25 as GPIO

   PINSEL1= 0x00000000;

2. Configure PORT0.25 pin as output.

   IO0DIR= 0x02000000;

3. To turn ON the buzzer set P0.25 output high

Buzzer Program

1. Set P0.25 as GPIO

   PINSEL1= 0x00000000;

2. Configure PORT0.25 pin as output.

   IO0DIR= 0x02000000;

3. To turn ON the buzzer set P0.25 output high

   IO0SET = 0x02000000;

Buzzer Program

1. Set P0.25 as GPIO

   PINSEL1= 0x00000000;

2. Configure PORT0.25 pin as output.

   IO0DIR= 0x02000000;

3. To turn ON the buzzer set P0.25 output high

   IO0SET = 0x02000000;

4. To turn OFF the buzzer set P0.25 output low

Buzzer Program

1. Set P0.25 as GPIO

   PINSEL1= 0x00000000;

2. Configure PORT0.25 pin as output.

   IO0DIR= 0x02000000;

3. To turn ON the buzzer set P0.25 output high

   IO0SET = 0x02000000;

4. To turn OFF the buzzer set P0.25 output low

Buzzer Program

1. Set P0.25 as GPIO

   PINSEL1= 0x00000000;

2. Configure PORT0.25 pin as output.

   IO0DIR= 0x02000000;

3. To turn ON the buzzer set P0.25 output high

   IO0SET = 0x02000000;

4. To turn OFF the buzzer set P0.25 output low

   IO0CLR = 0x02000000;

Buzzer Program

1. Set P0.25 as GPIO

   PINSEL1= 0x00000000;

2. Configure PORT0.25 pin as output.

   IO0DIR= 0x02000000;

3. To turn ON the buzzer set P0.25 output high

   IO0SET = 0x02000000;

4. To turn OFF the buzzer set P0.25 output low

   IO0CLR = 0x02000000;

ARM Programming Tools

ARM Programming Tools

1. Software Required.

ARM Programming Tools

1. Software Required.

ARM Programming Tools

1. Software Required.

   Keil uVision4

ARM Programming Tools

1. Software Required.

   Keil uVision4

- Integrated Development Environment (IDE)
- Supports Developing and Debugging of ARM based microcontroller application
- Download Link: `http://www.keil.com/arm/mdk.asp`

ARM Programming Tools

1. Software Required.

   Keil uVision4

   - Integrated Development Environment (IDE)
   - Supports Developing and Debugging of ARM based microcontroller application
   - Download Link: `http://www.keil.com/arm/mdk.asp`

2. Hardware Required

ARM Programming Tools

1. Software Required.

   Keil uVision4

   - Integrated Development Environment (IDE)
   - Supports Developing and Debugging of ARM based microcontroller application
   - Download Link: `http://www.keil.com/arm/mdk.asp`

2. Hardware Required

ARM Programming Tools

1. Software Required.

   Keil uVision4

   - Integrated Development Environment (IDE)
   - Supports Developing and Debugging of ARM based microcontroller application
   - Download Link: `http://www.keil.com/arm/mdk.asp`

2. Hardware Required

   - hex file can be loaded into microcontroller using

ARM Programming Tools

1. Software Required.

   Keil uVision4

   - Integrated Development Environment (IDE)
   - Supports Developing and Debugging of ARM based microcontroller application
   - Download Link: `http://www.keil.com/arm/mdk.asp`

2. Hardware Required

   - hex file can be loaded into microcontroller using

     a. BootLoader

ARM Programming Tools

1. Software Required.

   Keil uVision4

   - Integrated Development Environment (IDE)
   - Supports Developing and Debugging of ARM based microcontroller application
   - Download Link: `http://www.keil.com/arm/mdk.asp`

2. Hardware Required

   - hex file can be loaded into microcontroller using
     a. BootLoader

ARM Programming Tools

1. Software Required.

   Keil uVision4

   - Integrated Development Environment (IDE)
   - Supports Developing and Debugging of ARM based microcontroller application
   - Download Link: `http://www.keil.com/arm/mdk.asp`

2. Hardware Required

   - hex file can be loaded into microcontroller using
     a. BootLoader
     b. ARM Programmers viz. FlashMagic, LPC2000 Flash Utility, winARM etc..

Syntax for C-Program

Syntax for C-Program

#include

Syntax for C-Program

#include

```
#include <lpc214x.h>
```

Syntax for C-Program

#include

```
#include <lpc214x.h>
```

Pin Configuration

Syntax for C-Program

#include

```
#include <lpc214x.h>
```

Pin Configuration

```
void Init_buzzer_pin (void)
{
      PINSEL1 =
      IO0DIR =
      IO0CLR =   \\ Initially buzzer off
}
```

Syntax for C-Program

Syntax for C-Program

Main-Program

Syntax for C-Program

Main-Program

```
int main (void)
{
   Init_buzzer_pin ();
   while(1)
    {
      buzzer_on();
      delay_ms();
      buzzer_off();
      delay_ms();
    }
}
```

Syntax for C-Program

Main-Program

```
int main (void)
{
   Init_buzzer_pin ();
   while(1)
    {
      buzzer_on();
      delay_ms();
      buzzer_off();
      delay_ms();
    }
}
```

Functions

Syntax for C-Program

Main-Program

```
int main (void)
{
   Init_buzzer_pin ();
   while(1)
    {
      buzzer_on();
      delay_ms();
      buzzer_off();
      delay_ms();
    }
}
```

Functions

```
void buzzer_on (void)
{
     IO0SET = 0x02000000;
}
```

Syntax for C-Program

Main-Program

```
int main (void)
{
   Init_buzzer_pin ();
   while(1)
    {
      buzzer_on();
      delay_ms();
      buzzer_off();
      delay_ms();
    }
}
```

Functions

```
void buzzer_on (void)
{
     IO0SET = 0x02000000;
}
```

```
void buzzer_off (void)
{
     IO0CLR = 0x02000000;
}
```

Thank You!

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