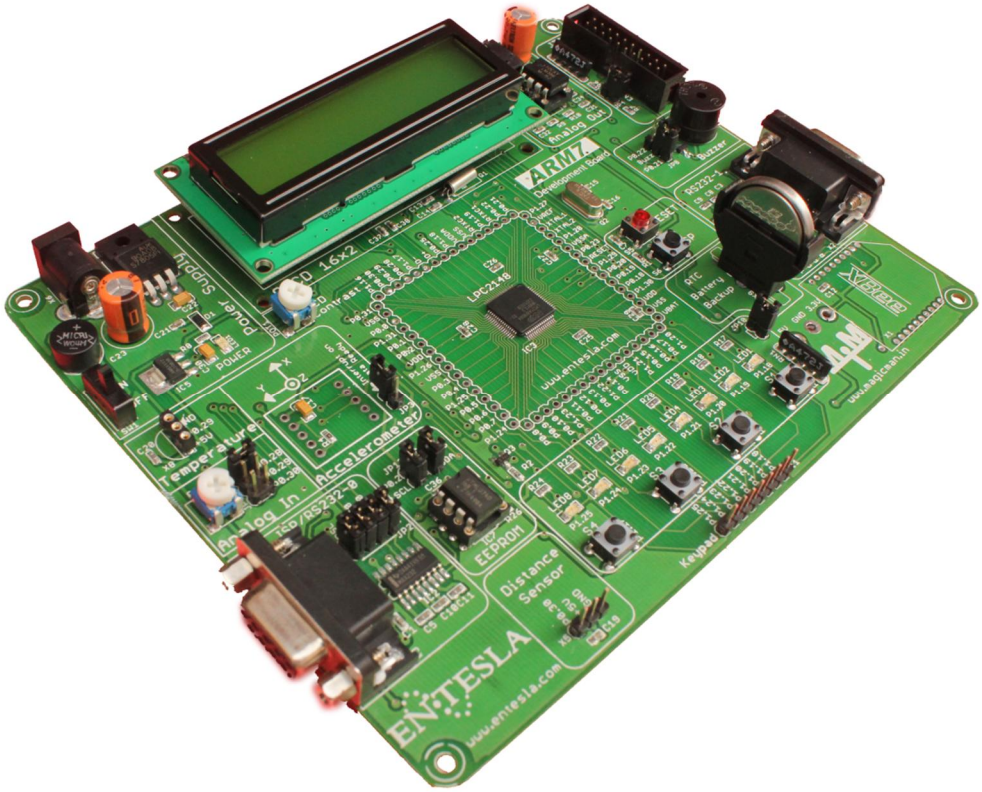
ARM7 Development board

USER MANUAL



A product of ENTESLA

Features

1. An on-board NXP's LPC2148 microcontroller with ARM7 architecture.
2. A power jack for an external power supply from 8 to 16V AC/DC. Alternatively the board can be powered through the on-board USB connector.
3. A 16x2 Character LCD interface operating in 4-bit mode. A potentiometer to vary the contrast of Character LCD.
4. A 3.5mm audio jack to connect the D/A output of microcontroller to a speaker or headphones.
5. Standard JTAG connector to support programming/debugging of LPC2148 microcontroller.
6. An on-board crystal of 12MHz to clock the microcontroller to upto 60MHz with internal PLL.
7. A push button to hardware reset the MCU.
8. A push button used in combination with RESET button to enter ISP mode.
9. On-board buzzer with selectable input to generate various audio frequencies.
10. Two DB9 connectors to implement RS232 communication using the on-chip UART0 and UART1.
11. An on-board crystal of 32.768kHz and a battery backup for on-chip Real time Clock.
12. A socket to plug-in Digi's 2.4GHz XBEE RF transceiver.
13. Four Push Buttons to provide digital input to selected GPIO pins of the microcontroller.
14. Eight on-board user programmable LEDs on selected GPIO pins with an option to disable them using a jumper.
15. A connector for 4x4 Keypad interface.
16. A connector to plug-in distance measurement sensors such as Sharp's GP2DXX or Maxbotic's Ultrasonic Sensors.
17. An I2C based EEPROM.
18. The microcontroller can be programmed using the on-chip serial bootloader using the RS232-0 port.
19. A socket to plug-in SPI based 3-axis accelerometer.
20. Variable Analog Input using potentiometer using one of the three user selectable channels.
21. Socket for interfacing temperature sensor such as Microchip's MCP9700.
22. On-board USB connector interfaced to on-chip USB peripheral of LPC2148 for USB based communication.
23. A Breakout for all the pins present on the microcontroller.
24. Well-documented top and bottom side for easy hardware troubleshooting.

Copyright Notice

Disclaimer

Reasonable care has been taken to prepare this document, but no guarantee is given to correctness of its content. Our products are under continual improvement and we reserve the rights to make changes without any notice.

Trademark

All trademarks are properties of their respective owners.

NXP® is a registered trademark of NXP Semiconductors.

Revision History

Revision	Revision History	Date
v1.1	First Release	May 2011

Table of Contents

Package Contents

- ARM7 development board.
- Product CD-ROM with demo codes with complete source code for all the peripherals and additional material for getting started with ARM7 on LPC2148 microcontroller.
- A Temperature Sensor (MCP9700) #.
- A MMA7455 Accelerometer Breakout #.
- 12V AC/DC adapter as external power supply#.
- RS232 Cable, USB Cable.
- Character LCD.

Getting Started

1. The ARM7 Development Board ships with the following configuration
    - An LPC2148 microcontroller (soldered).
    - A 12MHz Crystal.
    - All jumpers in default position. The default position is marked by a bracketed white line set against the jumper position as shown in **Figure.1**.

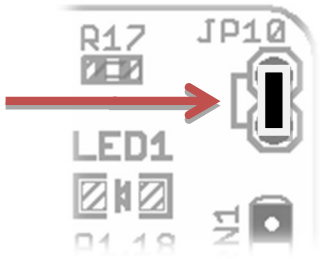


Figure.1

2. Plug in an external adapter and power the board by using the ON/OFF switch SW1. Optionally the board can also be powered using a USB cable connected to a computer.
3. The microcontroller shipped with the development board comes pre-programmed with an LED demo code. When board is powered ON, the user LEDs on-board blinks.
4. Explore the product CD which contains demo codes and various development tools.

 # Optional.

Introduction

Firstly we would like to thank you for your interest in ARM7 development board. We here at ENTESLA believe in providing you quality development tools for realizing your innovative ideas in the field of embedded systems.

ARM7 development board is an embedded system development board for NXP's famous LPC2148 microcontroller. It features various interfaces available on-board to unleash the power of the ARM7 architecture. It has been designed to help professionals, students and amateurs to explore the capabilities of ARM7 architecture using LPC2148 microcontroller and practice application development for various interfaces with minimal hardware reconfiguration. The ARM7 Development board comes with demos and source codes for getting started with LPC2148 microcontroller. The user is free to use the libraries provided with the board and make changes suiting their needs.

Safety Instructions

- Read the user manual carefully for exact jumper and switch position.
- Keep the development board away from humid environment.
- Ensure that you supply the board with 9 to 16 Volts AC/DC.
- Always power off the board before inserting sensors or LCD.
- All cautions and warnings should be noted.

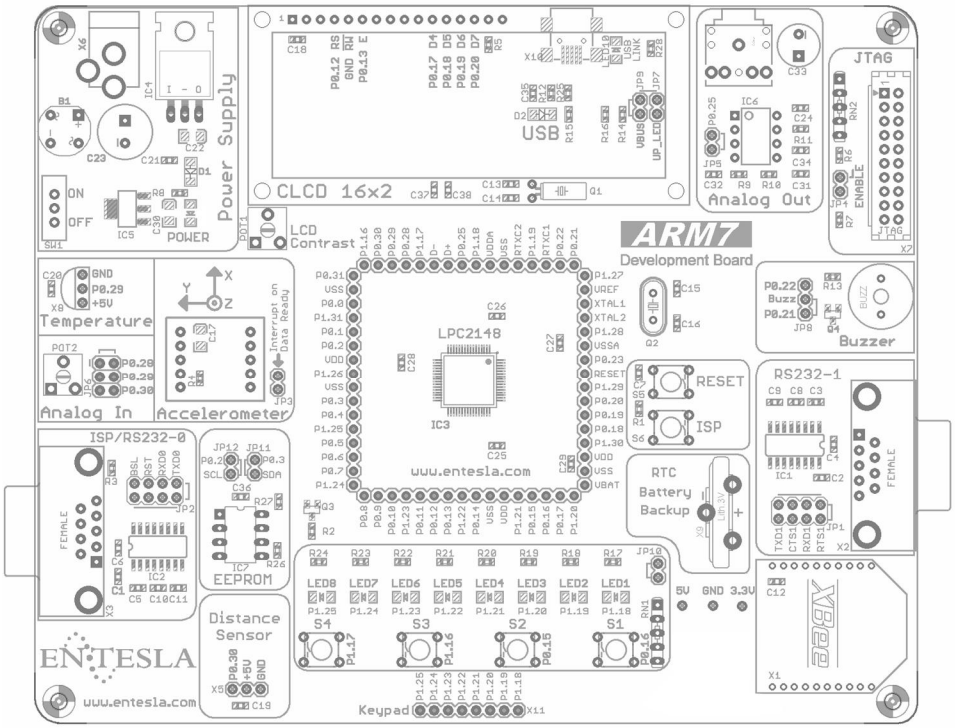


Figure.2

Power Supply

The ARM7 development board can be powered by either an external power supply or using the USB.For external power supply, ensure that the output is in range of 8 to 16 Volts AC/DC with a minimum current rating of 200 mA. An onboard LM7805 voltage regulator is used to regulate the AC/DC input to +5V output. Electrolytic capacitors are used to reduce the ripples in the voltage. The ceramic capacitors are used to suppress the high frequency noise present at the output of the voltage regulator. A 3.3V regulator is also provided for powering the microcontroller and other 3.3V operable devices on-board.

In absence of external power supply, the board can also be powered through the USB using a computer's USB port. An ON/OFF switch (SW1) is provided to control the external power supply and an LED is used as an indicator.

Figure.3 illustrates the schematic of the power supply section on the ARM7 development board.

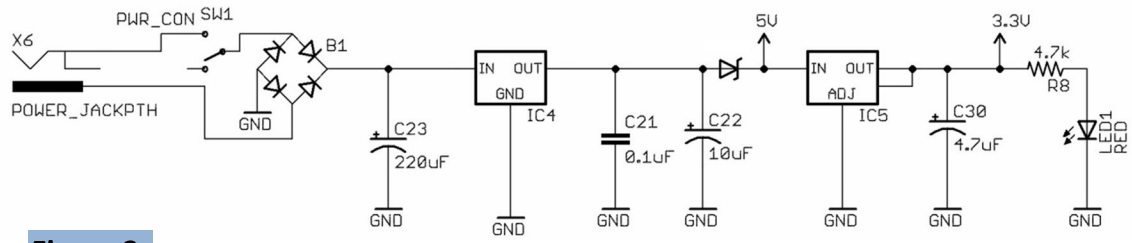


Figure.3

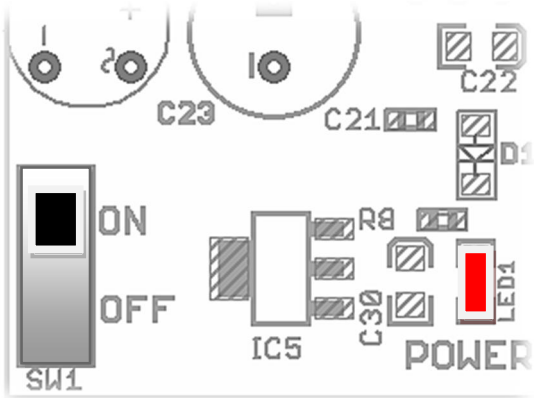


Figure.4

Note: **The power control switch works only with the external power supply and has no use when powered using the USB.**

Microcontroller and Oscillator Circuit

The ARM7 development board features NXP's LPC2148 microcontroller. It is based on 16-bit/32-bit ARM7TDMI-S CPU Architecture. It features 32kB of RAM and 512kB of on-chip flash ROM. The 128-bit wide interface/accelerator enables high speed 60MHz operation. The **Figure.5** below illustrates the microcontroller and the oscillator circuit.

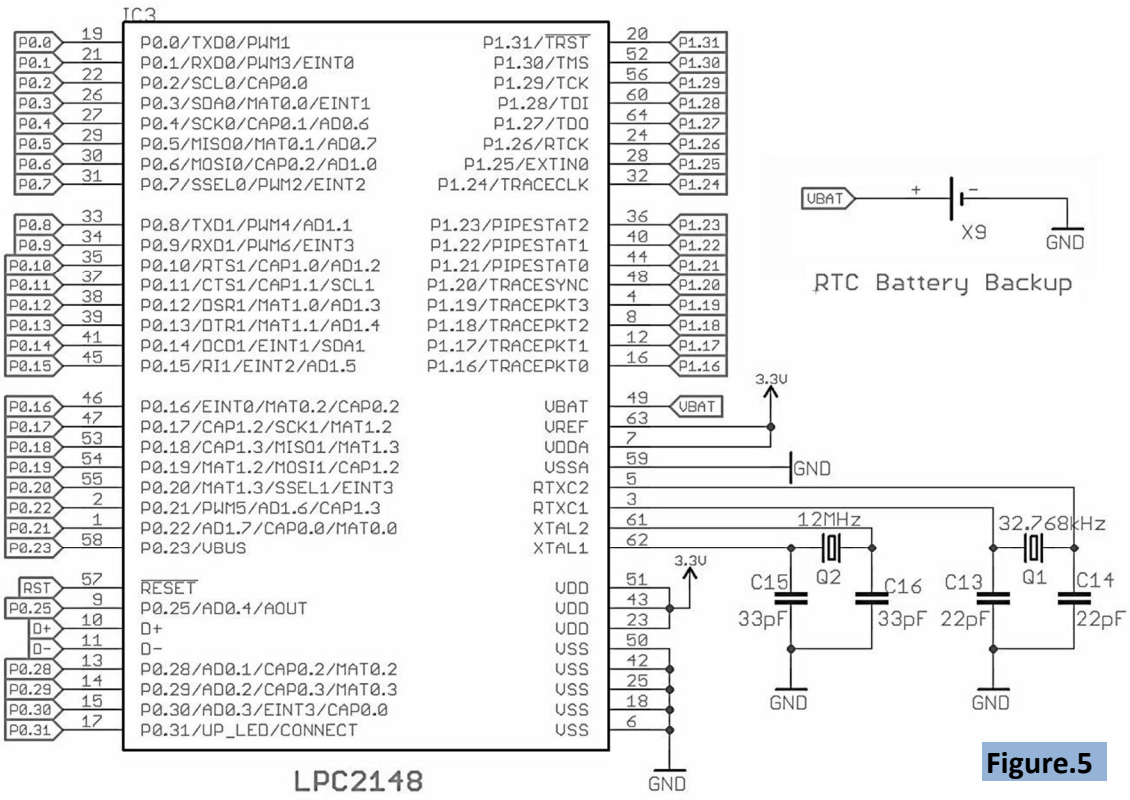
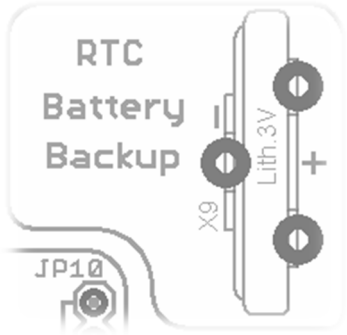


Figure.5

The development board comes with a 12MHz crytal oscillator soldered. Using the internal PLL block of LPC2148, the processor can be clocked to upto 60MHz. LPC2148 features an on-chip Real Time Clock. A 32.768kHz crystal provides the clock required to run the RTC. A battery backup of 3V is provided using CR2032 battery to keep the Real Time Clock ticking even in absence of external power.



Note: **Ensure the battery is inserted in the right manner into the battery hoder. Reversing the polarity may damage the microcontroller permanently.**

Reset

A reset is generated by holding the RESET pin low. As shown in **Figure.6.** Reset button can be used to reset the microcontroller. On reset all the pins of PORT0 and PORT1 are configured as input with an exception when JTAG and TRACE functionality is enabled. Refer datasheet of LPC2148 for more information.

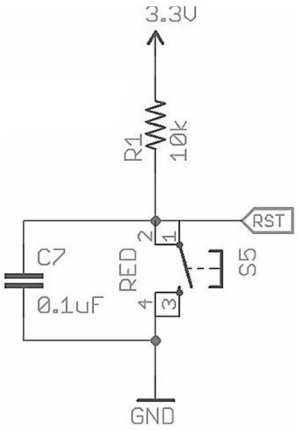


Figure.6

Programmer/Debugger Interface

The ARM7 Development board provides an option to program the microcontroller on-board using the on-chip serial bootloader or using the JTAG interface.

The RS232-0 port can be used to download program code into the internal Flash ROM of the LPC2148 microcontroller. To accomplish this ISP (In-System Programming) via a bootloader and an In-Application Programming feature of the LPC2148 microcontroller is used. The standard way to start ISP is to hold P0.14 low while issuing a processor reset. This can be manually done using the RESET and ISP push button. The procedure can be automated by setting the BSL and RST jumpers on jumper group JP2. The RESET signal is controlled by the DTR serial signal and P0.14 using th RTS serial signal. The tool Flash Magic from ES Academy is recommended to download the program code to the internal Flash. The **Figure.7** below shows the jumper setting for the automating the ISP and **Figure.8** shows the jumper setting to use the on-board ISP push buttons. Refer the schematic in RS232-0 section for more information.

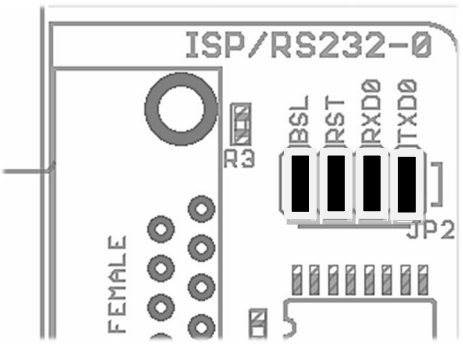


Figure.7

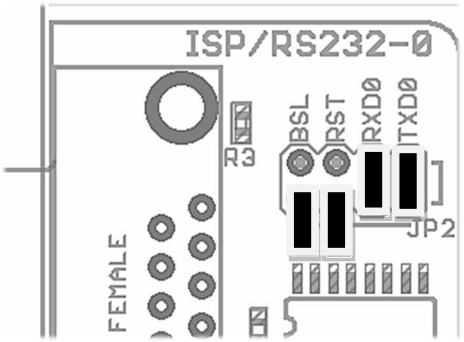
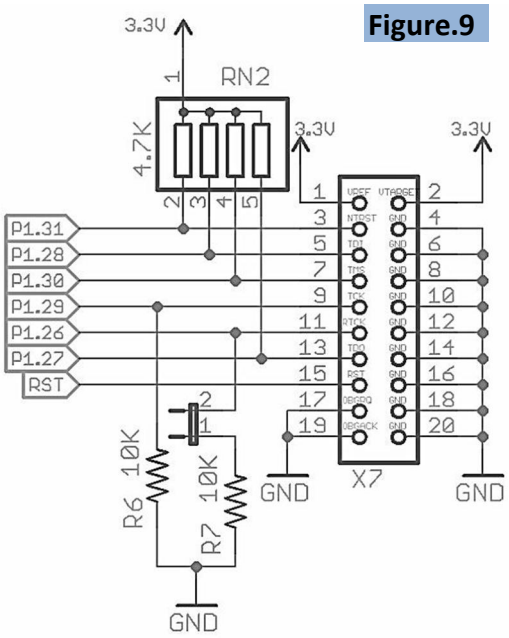


Figure.8

JTAG interface for programming and debugging is provided through a standard header of 20-pin. The interface shares 6 GPIO lines and is enabled by pulling line P1.26 low during reset using jumper JP4. The **Figure.9** shows the schematic for the JTAG interface. The **Figure.10** shows the jumper position required to enable the JTAG functionality.

Figure.9



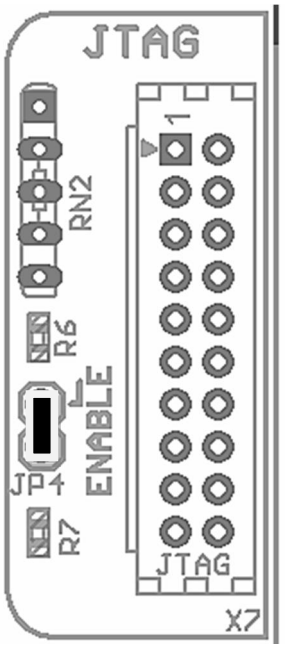


Figure.10

LEDs

There are a total of 8 LEDs connected to the port pins of the microcontroller on ARM7 Development Board. The LEDs are connected to PORT pins of P1.18 to P1.25. The LEDs come in color of RED, Orange and GREEN, the user can use them creatively as required. The jumper JP10 can be used to enable or disable the LEDs on the development board. The schematic of the LED interface is shown in **Figure.12**. The **Figure.11** below shows the jumper position to enable the LEDs.

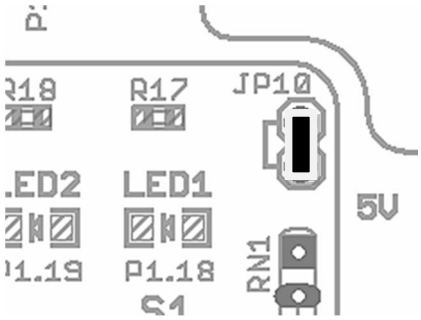


Figure.11

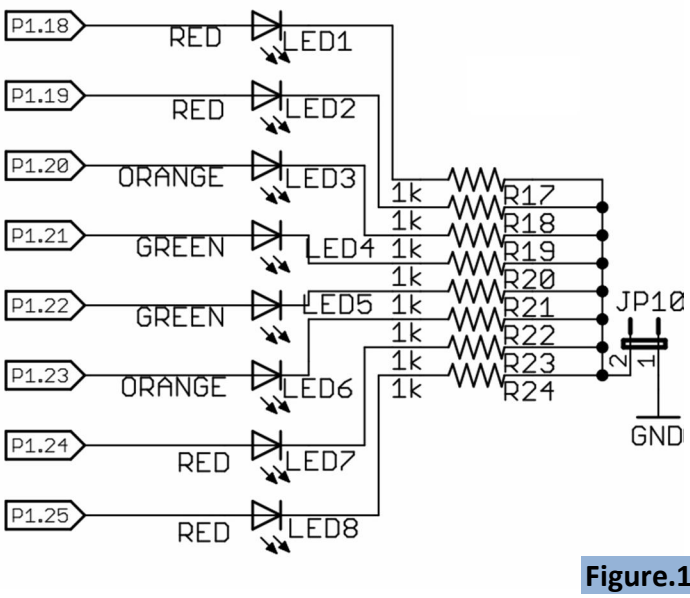


Figure.12

Note: **The LEDs are multiplexed with the Keypad. The user can disable the LEDs while using the keypad.**

Push Buttons

The ARM7 Development board has a total of 4 Push Buttons connected to port pins P0.16, P0.15, P1.16, P1.17 of the microcontroller. The port pins to which the push buttons are connected are pulled down to define a Logic '0' when button is not pressed. A Logic '1' is applied to the microcontroller when the push button is pressed. The **Figure.13** below shows the schematic for push buttons interfaced to the microcontroller.

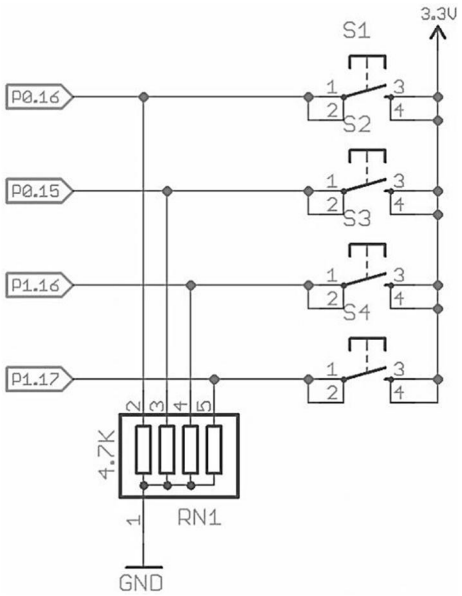


Figure.13

FYI: **Push button S1 and S2 are connected to EINT0 and EINT2 pins of the microcontroller.**

Analog In

The LPC2148 has 14 Analog channels connected to two different 10-bit Analog to Digital Converters. A Potentiometer POT1 provides the analog input to the microcontroller. User has an option to connect the analog input to one of the P0.28/AN0.1, P0.29/AN0.2, P0.30/AN0.3 channels. The **Figure.14** below shows the jumper position to connect analog input to P0.28/AN0.1 channel. The schematic of analog input is shown in **Figure.15**.

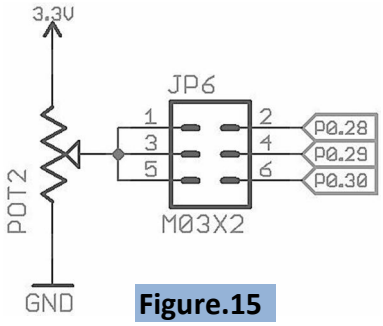


Figure.15

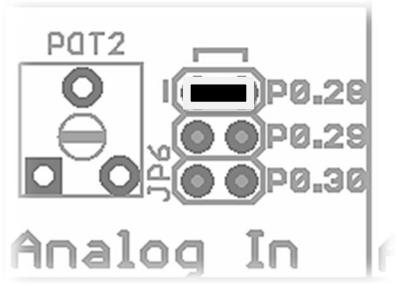


Figure.14

Note: **The channels P0.29/AN0.2 and P0.30/AN0.3 are shared with temperature sensor and distance sensor respectively. Ensure to disconnect the jumper from these positions when using the respective sensor.**

Analog Out

The LPC2148 microcontroller has one analog output, available as an alternative function on pin P0.25. A power amplifier LM386 is provided to convert the analog out signal to sound. A 3.5mm audio out jack is provided to connect a headphone or a speaker. The **Figure.16** shows the jumper position to connect the analog out to the power amplifier circuit. The schematic of the Analog Out section is shown in **Figure.17**.

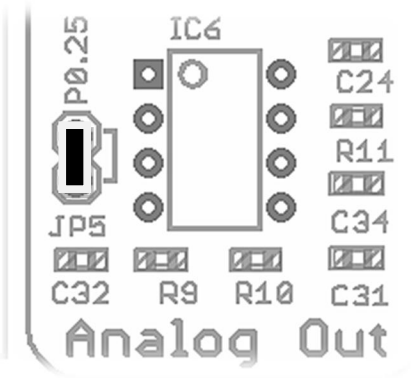


Figure.16

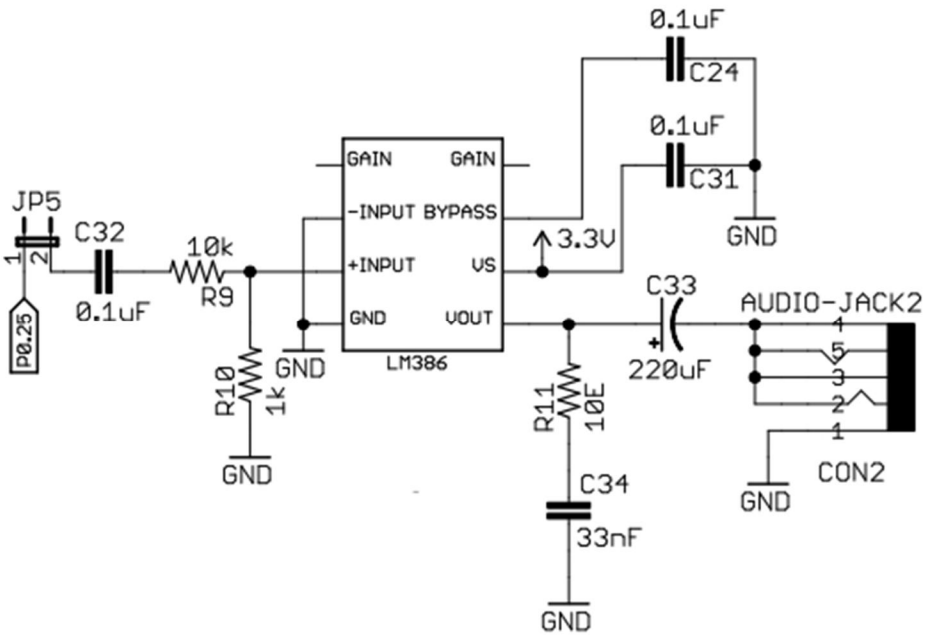


Figure.17

Character LCD (4-bit Mode)

ARM7 Development board provides a connector to interface a Character LCD of 16 columns and 2 rows. Each character is made up of 5x7 pixels. The data lines of the CLCD are connected to pins of PORTD and control lines to port pins P0.17 to P0.20. The **Figure.18** illustrates the schematic for operating the LCD in 4-bit mode.

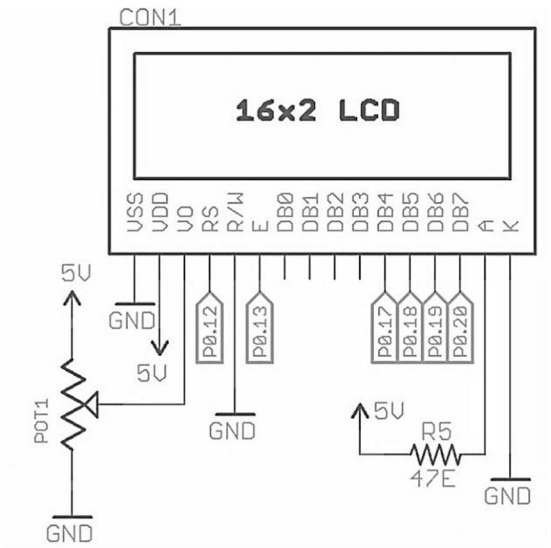


Figure.18

A potentiometer POT1 is provided to manually control the contrast of the LCD.

Figure.19

Note: **Do not plug the CLCD into its connector when power is switched ON. It may damage the CLCD permanently.**

Electronic Buzzer

A 12mm round electronic buzzer is provided on-board. It operates around the audible range of 2 kHz. An NPN transistor circuit drives the buzzer to generate sounds of varying frequency. The base of the buzzer is connected to a port pin. User has an option to connect either P0.21 or P0.22 using jumper JP8. The **Figure.20** shows the jumper placement to connect P0.21 to the base of the transistor.

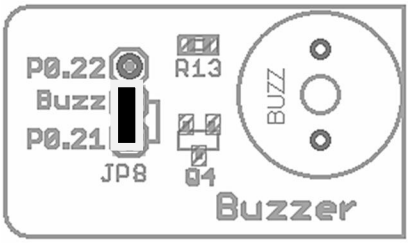


Figure.20

FYI: **P0.21 is a PWM channel. Signals of varying frequency can easily be generated using this channel while your processor is processing another task.**

RS232-0 Interface

The LPC2148 microcontroller features dual UART. The UART0 interface is available to the user using the RS232-0 port and UART1 is available using the RS232-1 port on the ARM7 development board.

Each RS232 circuit comes with a MAX232, voltage level shifter to convert the TTL signals of the microcontroller to RS232 level. The RS232-0 port can be used as a standard communication port for serial communication between the microcontroller and the computer. **Figure.21** shows the jumper position to establish a standard serial communication.

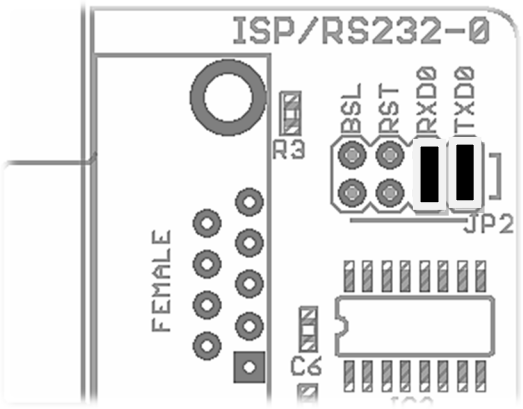


Figure.21

The RS232-0 port can also be used to download program code into the internal Flash ROM of the LPC2148 microcontroller using ISP. Refer the Programming/Debugging interface section for more information about ISP using RS232-0. The **Figure.22** below shows the schematic of RS232-0 interface.

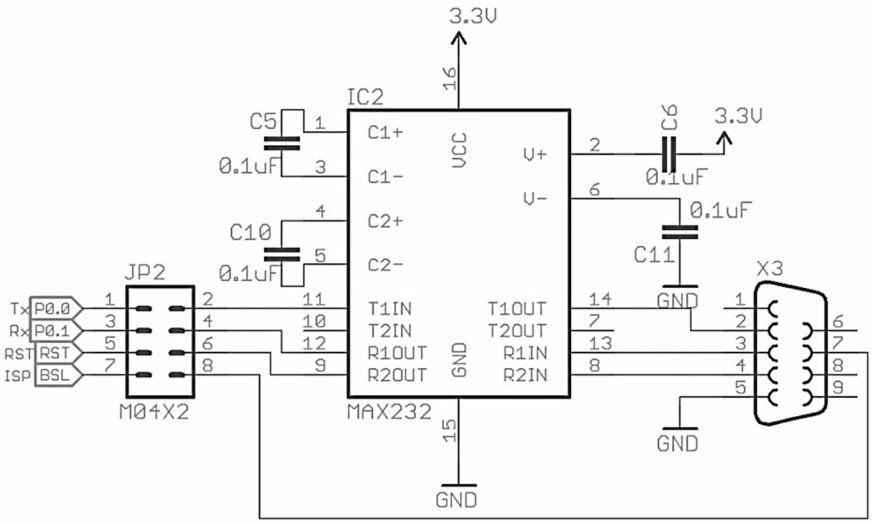
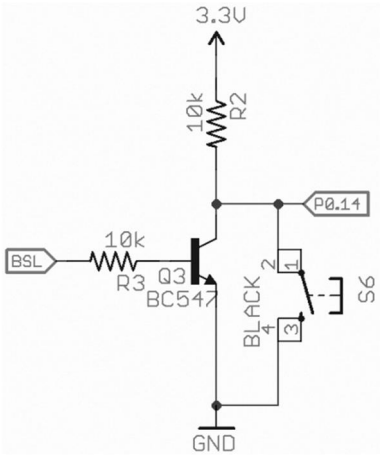


Figure.22



RS232-1 Interface

The LPC2148 microcontroller features dual UART. The UART0 interface is available to the user using the RS232-0 port and UART1 is available using the RS232-1 port on the ARM7 development board.

The RS232 circuit comes with a MAX232, voltage level shifter to convert the TTL signals of the microcontroller to RS232 level. The RS232-1 port can be used as a standard communication port for serial communication. RTS and CTS lines are also provided for this port to implement hardware flow control**. Figure.23** shows the jumper position to establish a serial communication with hardware flow control. Refer **Figure.24** for the schematic of RS232-1 Interface.

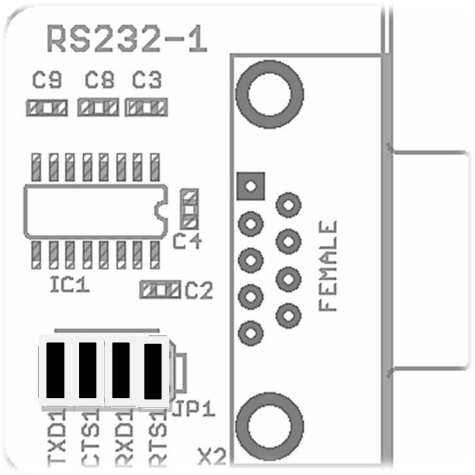


Figure.23

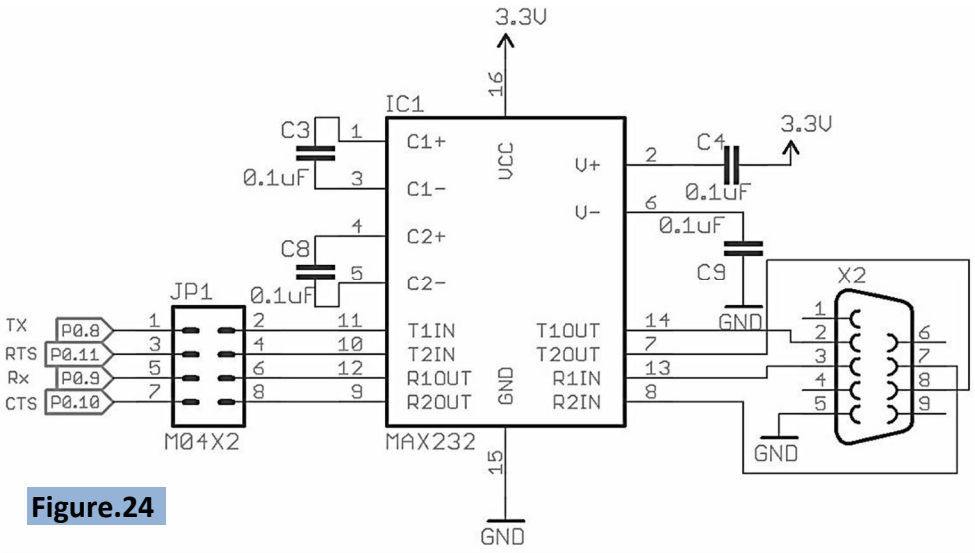


Figure.24

USB

The LPC2148 features an on-chip USB2.0 complaint peripheral. ARM7 development board provides a USB connector to make the best of this feature. The USB birectional D+ and D- lines are hardwired to the connector. A USB good link indicator LED is provided to show the status of the USB link. An option to detect the presence of USB power is provided using the VBUS pin. **Figure.25** shows the jumper placement to enable the link indicator and USB power detection feature. If disabled by disconnecting the jumper, these port pins can be used as GPIO pins. The schematic of the USB interface is shown in **Figure.26**.

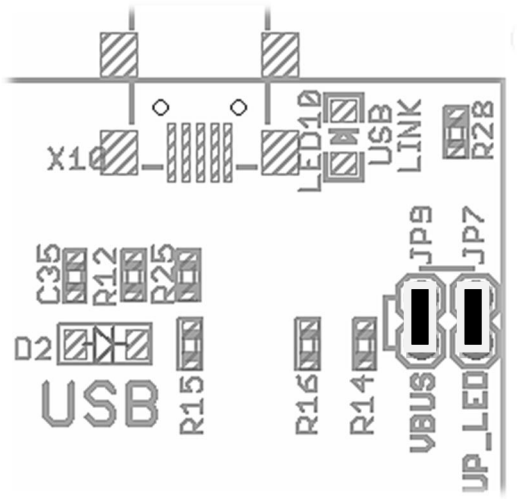


Figure.25

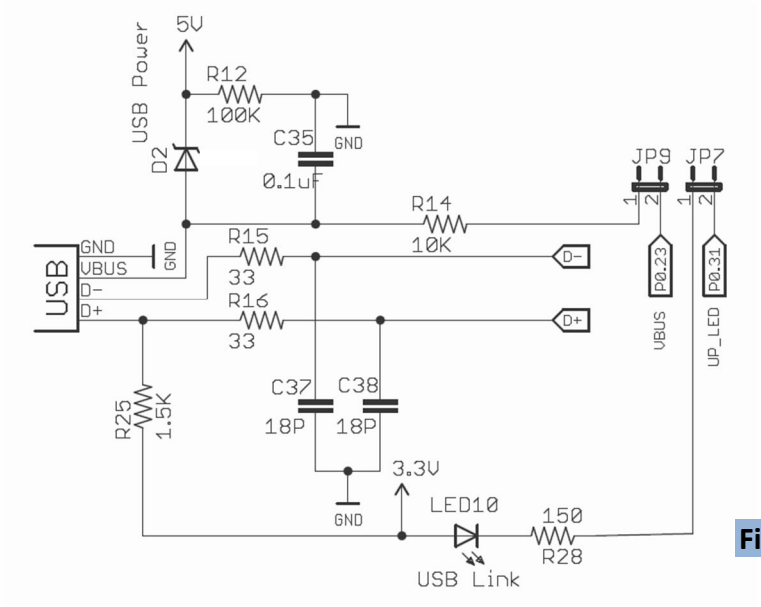


Figure.26

FYI: **In absence of an external power source, the board can be powered using the USB port available on a computer.**

EEPROM

The LPC2148 microcontroller does not have an on-chip EEPROM for data that must be saved when power is removed. The ARM7 development board provides Atmel's AT24C04 EEPROM for storing small amount of data like calibration tables and data logs. I2C protocol is used to communicate with the EEPROM. **Figure.27.** shows the jumper placement to connect the EEPROM to the I2C0 lines of the LPC2148 microcontroller. The schematic of the the EEPROM interface is shown in **Figure.28.**

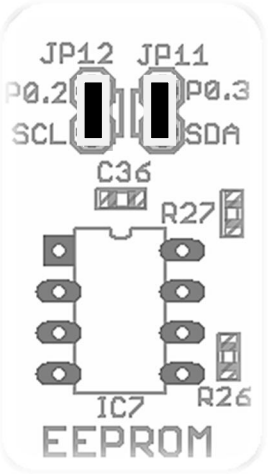


Figure.27

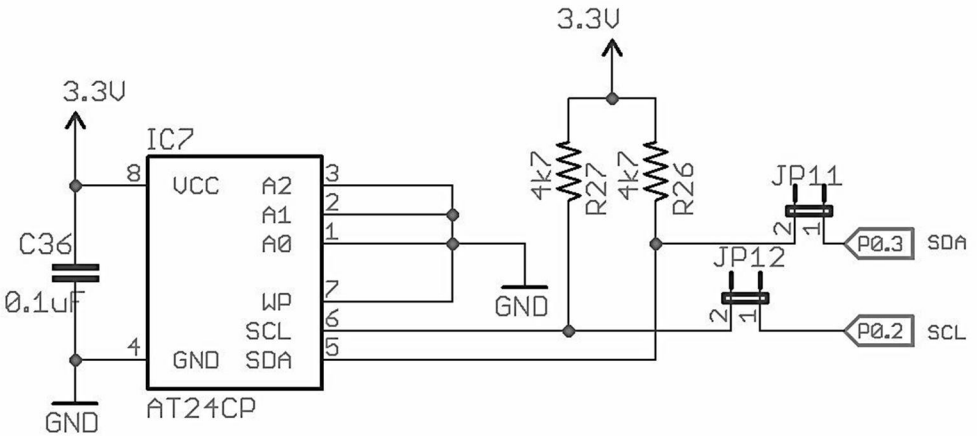


Figure.28

Temperature Sensor

Analog temperature sensors like LM35 or Microchip's MCP9700 can be directly plugged into the socket provided on-board for measuring temperature. **Figure.29.** shows the schematic of temperature sensor interfacing with the LPC2148 microcontroller.

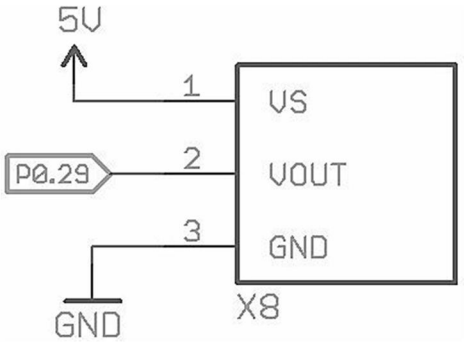


Figure.29

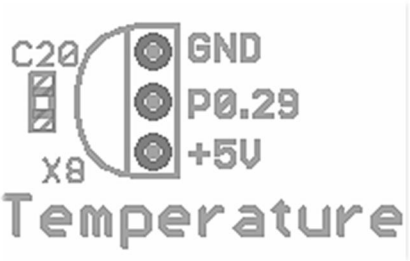


Figure.30

Note: **P0.29 is also shared with the Analog In section. Ensure that jumper JP6 is not used to select P0.29 as Analog Input.**

Note: **Improper placement of the temperature sensor may damage the sensor permanently.**

Distance Sensor

Analog distance sensors like Sharp's GP2DXXX and Maxbotic's Ultrasonic sensor can directly be interfaced with the ARM7 Development board for distance measurement from an obstacle. **Figure.31** shows the schematic of distance sensor interfacing with the LPC2148 microcontroller.

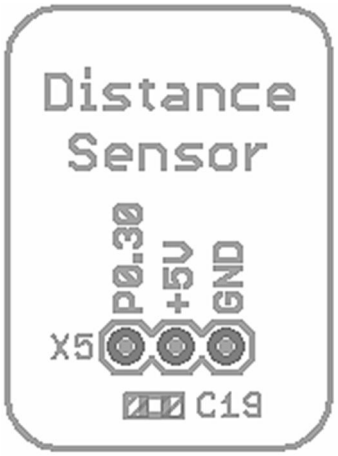


Figure.31

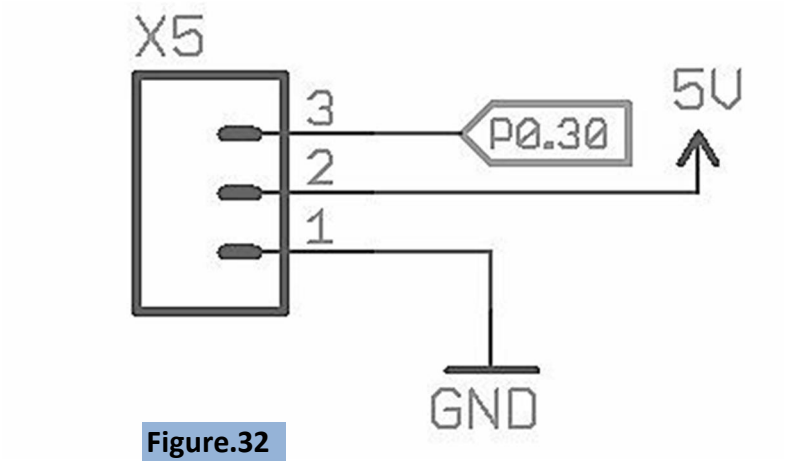


Figure.32

Note: **P0.30 is also shared with the Analog In section. Ensure that jumper JP6 is not used to select P0.30.**

Note: **Improper connection of the distance sensor to the development board may damage the sensor permanently.**

XBEE

A connector to add wireless ZIGBEE communication to your design is provided on the ARM7 Development board. Digi's XBEE and XBEE Pro modules are compatible with this connector. The XBEE devices communicate over UART1 of the LPC2148. The UART1 is also shared with the RS232-1 port. Ensure to disconnect the RXD1 and TXD1 jumpers in RS232-1 section to use UART1 for XBEE as shown in **Figure.33**.

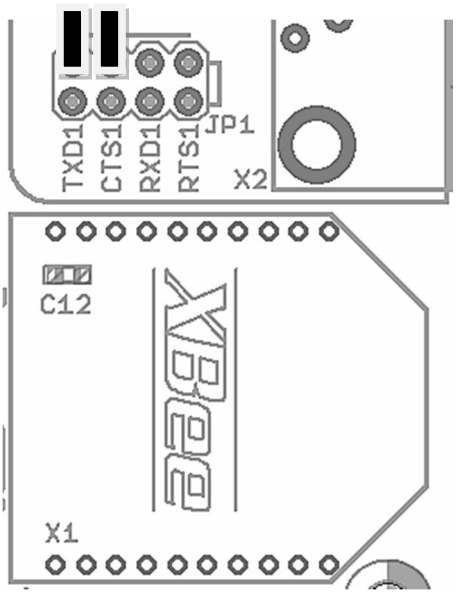


Figure.33

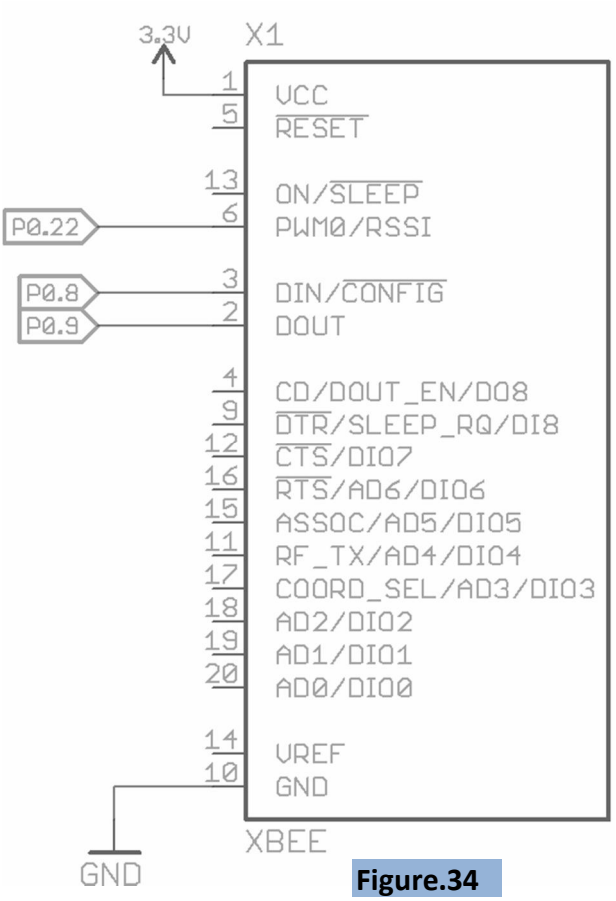


Figure.34

Note: **P0.22 is also shared with an optional buzzer port pin. When using RSSI feature of XBEE ensure that the P0.22 is disconnected as buzzer input.**

Accelerometer

A provision to add motion detection sensor to your design is possible on ARM7 development board. The breakout connector provided is directly compatible with Freescale's MMA7455 3-axis accelerometer. It communicates with the microcontroller using the SPI protocol. The **Figure.36** below shows the schematic and the jumper position to enable the interrupt on data ready feature of the device.

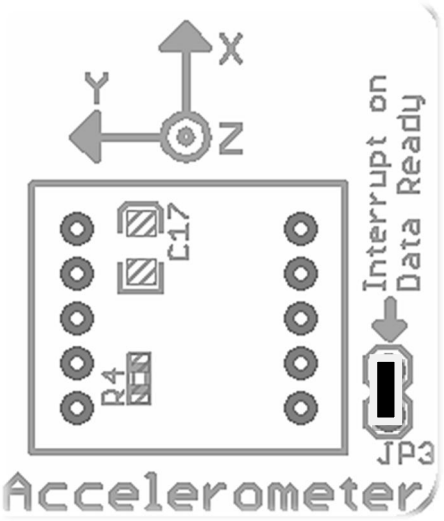


Figure.35

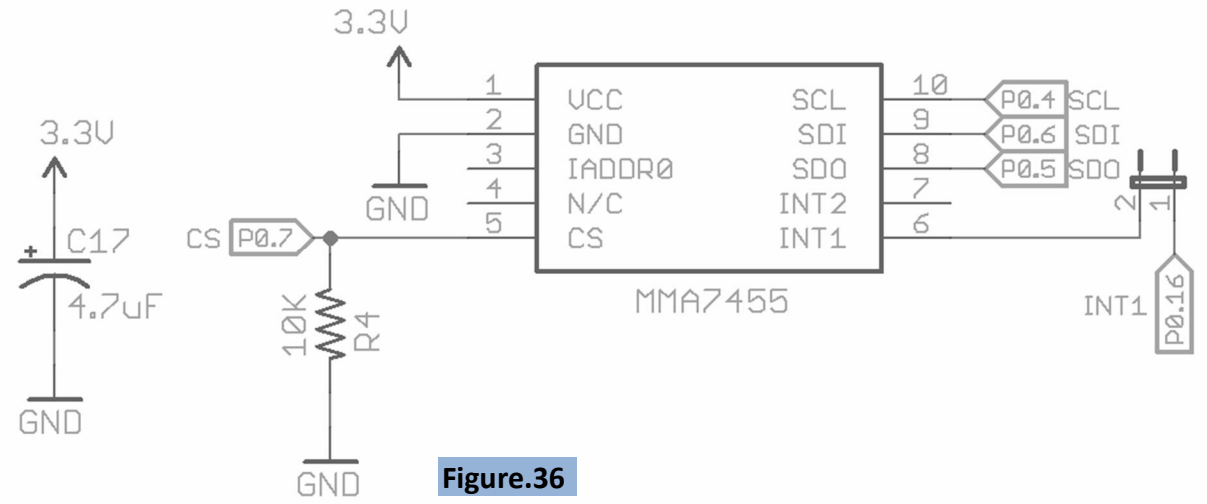


Figure.36

Note: **P0.16 is also shared with the push button S1. Avoid using push button S1 when using the interrupt on data ready feature.**

Keypad

A connector to interface a 4x4 keypad is provided at the bottom edge of the ARM7 development board. A total of 8 lines are required to interface a 4x4 keypad to a microcontroller. Four of the lines are connected to input pins and the other four to the output pins of the microcontroller. In this case the keypad is connected to port pins P1.18 to P1.25. These lines when defined as input are internally connected to a strong pull-up of 60k-300kΩ (refer the datasheet for more information). Therefore the need to connect external pull-ups to the input lines to define a Logic “HIGH” state on “no key press” is not necessary. The **Figure.37** below shows the schematic of the 4x4 keypad interface.

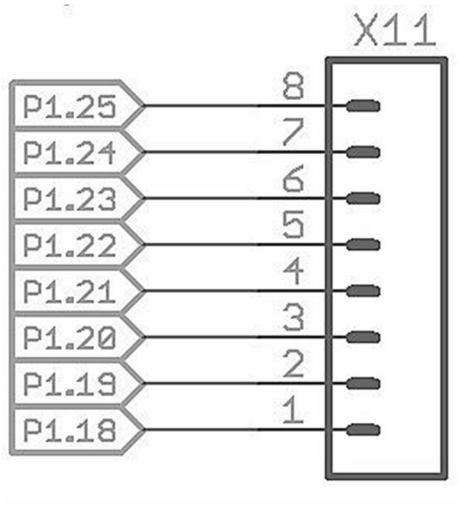


Figure.37

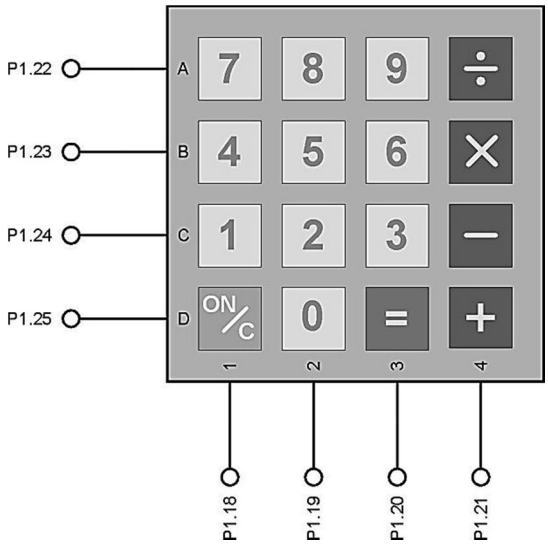


Figure.38

Note: **The port pins P1.25 to P1.18 are multiplexed with the on-board LEDs. Disable the LEDs using jumper J10 when using the Keypad.**

Notes

If a problem arises with the development board and no solution can be obtained from the user manual, please feel free to contact us at **techsupport@entesla.com**

For a feedback on the demo codes or user manual, email us at **feedback@entesla.com**

For business proposals or any other queries **contact@entesla.com**

Notes

If a problem arises with the development board and no solution can be obtained from the user manual, please feel free to contact us at **techsupport@entesla.com**

For a feedback on the demo codes or user manual, email us at **feedback@entesla.com**

For business proposals or any other queries **contact@entesla.com**

Embedded Designs and Solutions

www.entesla.com

If a problem arises with the development board and no solution can be obtained from the user manual, please feel free to contact us at **techsupport@entesla.com**

For a feedback on the demo codes or the user manual, email us at **feedback@entesla.com**

For business proposals or any other queries **contact@entesla.com**