**Chapter-1Preamble**

**1.1 Introduction**

In our proposed system we are using two ZigBee Transceiver based embedded system based on ARM microcontroller. One system is installed inside the vehicle and other system is installed at the entrance. When the vehicle enter at the entrance it was identified by the ZigBee Transceiver installed at the gate by verifying code of the vehicle ZigBeetransceiver.Then the ZigBee based system at the gate asks driver to enter password. This password is also verified by the system at the entrance gate. Once both the vehicle identification and the driver authentication has done then the system send the information to the security guard at the gate. According to this information the guard allows or denies access to the vehicle inside the premises.

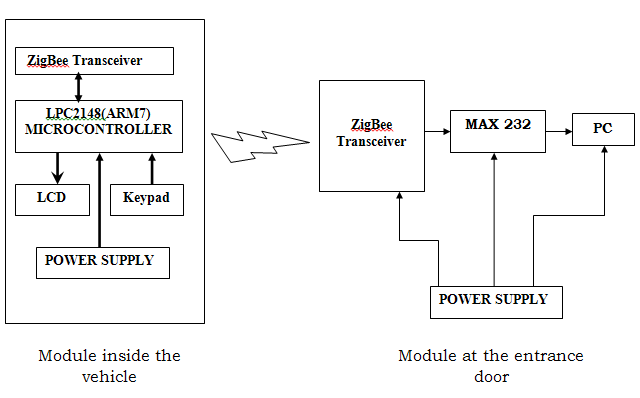


Figure 1.1.1: Block diagram of kit.

**1.2 Statement of the Problems**

1.2.1 Present model

* Manual security system in INDIA,specially in allvery important places like Parliament and VidhanSoudha.
* In the Parliament and VidhanSoudha,we have weak security system(manual security system) like if someone has red light vehicle or any government vehicle then he can easily enter in the parliament and that was the main reason whatever happened in the parliament like MILITANT attack.
* Figure 1.2.1.1: Present Scenario

1.2.2Remedies of present model

* AUTOMATIC ZIGBEE WIRLESS VEEHICULAR INDENTIFICATION AND AUTHENTICATION Security based systems.

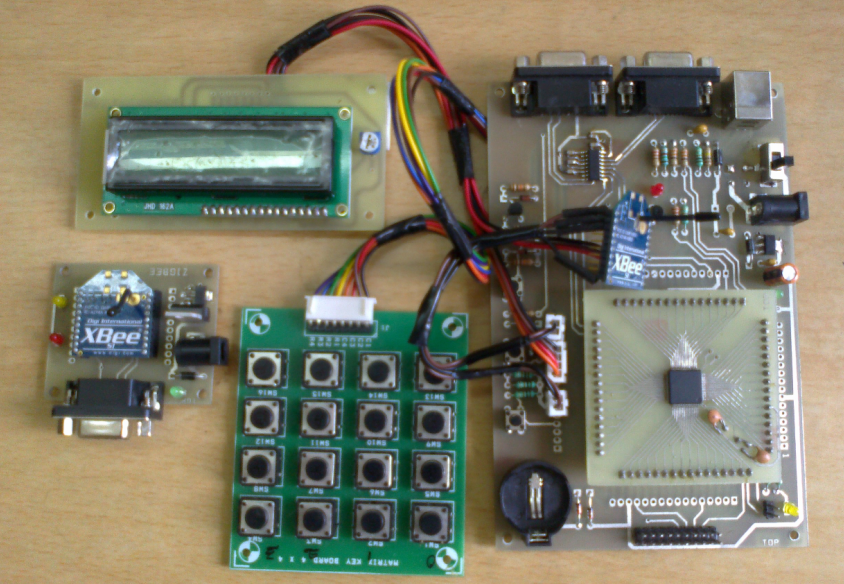


Figure 1.2.2.1: Remedy for Present Scenario

* There are two module in our present model(module1 and module2), now module 1 we will installed at the entrance of the parliament gate and module2 we will installed inside the vehicle.
* Now when the vehicle enter at the entrance it will identified by the ZIGBEE transceiver installed at the gate.
* After that the ZIGBEE based system at the gate asks the driver to enter the password and this password is also verified by the system at the entrance gate.
* Now once both the vehicle identification is and the driver authentication has been done then the information is being sent by the system to the security guard at the gate.
* According to this information the guard will allows or denies access to the vehicle inside the premises.

llllll

**1.3 Objective of study**

* Provide Security in all very important places as well as very important people through ZIGBEE WIRLESS VEEHICULAR INDENTIFICATION AND AUTHENTICATION SYSTEM.
* Use of ZigBee module ensures the system to be low cost and low power consumable system.
* User-friendly interface and simple system setup.

**1.4 Scope of study**

* Vehicles identification at VIPs places
* E.g PARLIAMENT,VIPs HOMES,HOTELs(all high level hotels) and apartments.

**1.5 Review of literature**

The 8051 Microcontroller and embedded systems by MAZIDI and LPC datasheets, and the user manual for the LPC2148.

**1.6: Methodology**

Three type of methodology we are using,i.e

* Software: Embedded C and visual basic.
* Tools:keil, Flash magic.
* TARGET DEVICE: LPC2148(ARM7)microcontroller.

**1.7 Limitation of study**

* Only one limitation
* we are using two module in our project that is module 1 and module2.
* Module1 we will installed at the entrance of the gate and module2 we will installed inside the vehicle.
* Now suppose if someone take module2 which we have installed inside the vehicle and he will installed the same module2 in his vehicle
* And take the worst case also, suppose he knows the password also then he can break the security system.

**2. HARDWARE DISCRIPTION**

2.1 LPC2148 MICROCONTROLLER

2.1.1 General Description

The LPC2141/42/44/46/48 microcontrollers are based on a 16-bit/32-bit ARM7TDMI-SCPU with real-time emulation and embedded trace support, that combine microcontroller with embedded high-speed flash memory ranging from 32 kB to 512 kB. A 128-bit widememory interface and a unique accelerator architecture enable 32-bit code execution atthe maximum clock rate. For critical code size applications, the alternative 16-bit Thumbmode reduces code by more than 30 % with minimal performance penalty.Due to their tiny size and low power consumption, LPC2141/42/44/46/48 are ideal forapplications where miniaturization is a key requirement, such as access control andpoint-of-sale. Serial communications interfaces ranging from a USB 2.0 Full-speed device,multiple UARTs, SPI, SSP to I2C-bus and on-chip SRAM of 8 kB up to 40 kB, make thesedevices very well suited for communication gateways and protocol converters, softmodems, voice recognition and low end imaging, providing both large buffer size and highprocessing power. Various 32-bit timers, single or dual 10-bit ADC(s), 10-bit DAC, PWMchannels and 45 fast GPIO lines with up to nine edge or level sensitive external interrupt pins make these microcontrollers suitable for industrial control and medical systems.

2.1.2 Features

* 16-bit/32-bit ARM7TDMI-S microcontroller in a tiny LQFP64 package.
* 8 kB to 40 kB of on-chip static RAM and 32 kB to 512 kB of on-chip flash memory.
* 128-bit wide interface/accelerator enables high-speed 60 MHz operation.
* In-System Programming/In-Application Programming (ISP/IAP) via on-chip bootloader software. Single flash sector or full chip erase in 400 ms and programming of256 bytes in 1 ms.
* EmbeddedICE RT and Embedded Trace interfaces offer real-time debugging with theon-chip RealMonitor software and high-speed tracing of instruction execution.
* USB 2.0 Full-speed compliant device controller with 2 kB of endpoint RAM.
* In addition, the LPC2146/48 provides 8 kB of on-chip RAM accessible to USB by DMA.
* One or two (LPC2141/42 vs. LPC2144/46/48) 10-bit ADCs provide a total of 6/14analog inputs, with conversion times as low as 2.44 ms per channel.
* Single 10-bit DAC provides variable analog output (LPC2142/44/46/48 only).
* Two 32-bit timers/external event counters (with four capture and four comparechannels each), PWM unit (six outputs) and watchdog.
* Low power Real-Time Clock (RTC) with independent power and 32 kHz clock input.
* Multiple serial interfaces including two UARTs (16C550), two Fast I2C-bus (400 kbit/s), SPI and SSP with buffering and variable data length capabilities.
* Vectored Interrupt Controller (VIC) with configurable priorities and vector addresses.
* Up to 45 of 5 V tolerant fast general purpose I/O pins in a tiny LQFP64 package.
* Up to 21 external interrupt pins available.
* 60 MHz maximum CPU clock available from programmable on-chip PLL with settling time of 100 ms.
* On-chip integrated oscillator operates with an external crystal from 1 MHz to 25 MHz.
* Power saving modes include Idle and Power-down.
* Individual enable/disable of peripheral functions as well as peripheral clock scaling for additional power optimization.
* Processor wake-up from Power-down mode via external interrupt or BOD.
* Single power supply chip with POR and BOD circuits: CPU operating voltage range of 3.0 V to 3.6 V (3.3 V ± 10 %) with 5 V tolerant I/O pads.

2.1.3 Block diagram of LPC2148

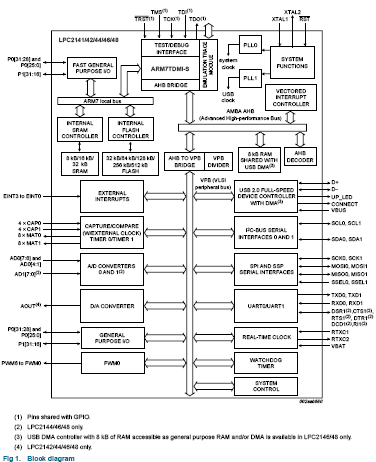
****

Figure 2.1.3.1: Lpc2148 block diagram

2.1.4 PIN DETAILS OF LPC2148

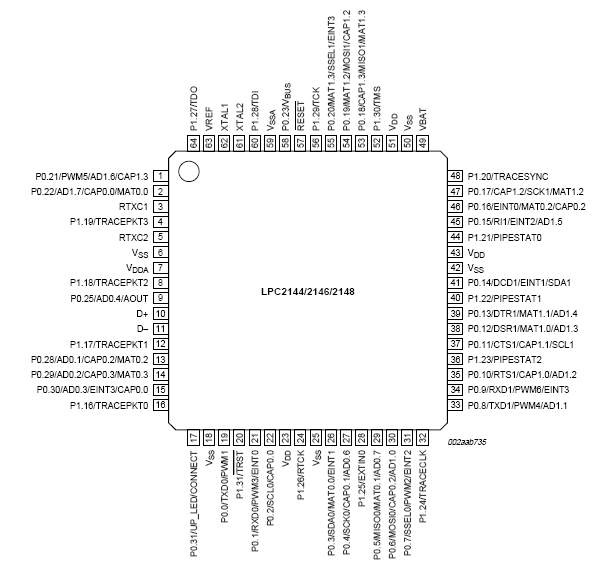
****

Figure 2.1.4.1: Lpc2148 pin details

**2.2 ZIGBEE**

****

Figure2.2.1:zigbeemodulesThe XBee and XBee-PRO OEM RF Modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices. The modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other.

2.2.1 Features

* Long Range Data Integrity
* Indoor/Urban: up to 100’ (30 m)
* Outdoor line-of-sight: up to 300’ (100 m)
* Transmit Power: 1 mW (0 dBm)
* Receiver Sensitivity: -92 dBm
* Low Power
* TX Current: 45 mA (@3.3 V)
* RX Current: 50 mA (@3.3 V)
* Power-down Current: < 10 μA
* ADC and I/O Line Support
* Analog-to-digital conversion, Digital I/OI/O Line Passing
* Advanced Networking & Security
* Retries and Acknowledgements
* DSSS (Direct Sequence Spread Spectrum)
* Each direct sequence channels has over 65,000 unique network addresses available
* Source/Destination Addressing
* Unicast & Broadcast Communications
* Point-to-point, point-to-multipoint and peer-to-peer topologies supported
* Coordinator/End Device operations
* Easy-to-Use
* configuration necessary for out-of box RF communications
* Free X-CTU Software (Testing and configuration software)
* AT and API Command Modes for configuring module parameters
* Extensive command set
* Small form factor

2.2.2 Mounting Considerations

The XBee/XBee-PRO RF Module was designed to mount into a receptacle (socket) and therefore does not require any soldering when mounting it to a board. The XBee Development Kits contain RS-232 and USB interface boards which use two 20-pin receptacles to receive modules.



Figure 2.2.2.1: Mounting Considerations

The receptacles used on MaxStream development boards are manufactured by Century Interconnect. Several other manufacturers provide comparable mounting solutions; however, MaxStream currently uses the following receptacles:

• Through-hole single-row receptacles - Samtec P/N: MMS-110-01-L-SV (or equivalent)

• Surface-mount double-row receptacles - Century Interconnect P/N: CPRMSL20-D-0-1 (or equivalent)

• Surface-mount single-row receptacles - Samtec P/N: SMM-110-02-SM-S

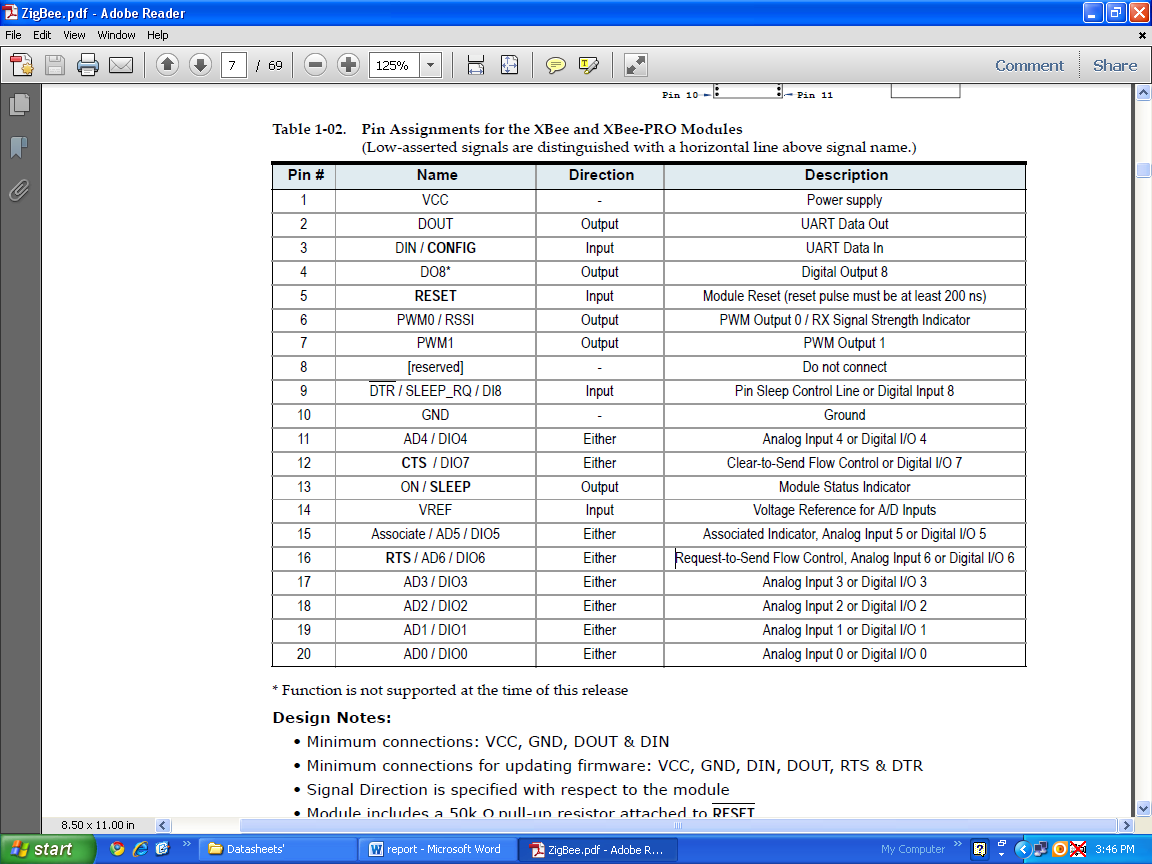
2.2.3 PinSignals 

Table: 2.2.3.1 pin signals of zigbee.

2.3 Dot Matrix Liquid Crystal Display Controller/Driver

2.3.1 Description

The HD44780U dot-matrix liquid crystal display controller and driver LSI displays alpha numeric, Japanese kana characters, and symbols. It can be configured to drive a dot-matrix liquid crystal display under the control of a 4- or 8-bit microprocessor. Since all the functions such as display RAM, character generator, and liquid crystal driver, required for driving a dot-matrix liquid crystal display are internally provided on one chip, a minimal system can be interfaced with this controller/driver. A single HD44780U can display up to one 8-character line or two 8 character lines. The HD44780U has pin function compatibility with the HD44780S which allows the user to easily replace an LCD-II with an HD44780U. The HD44780U character generator ROM is extended to generate 208 5 ´ 8 dot character fonts and 32 5 ´ 10 dot character fonts for a total of 240 different character fonts. The low power supply (2.7V to 5.5V) of the HD44780U is suitable for any portable battery-drivenproduct requiring low power dissipation.

2.3.2 Features

* + 5 ´ 8 and 5 ´ 10 dot matrix possible
  + Low power operation support:
    - 2.7 to 5.5V
  + Wide range of liquid crystal display driver power
    - 3.0 to 11V
  + Liquid crystal drive waveform
    - A (One line frequency AC waveform)
  + Correspond to high speed MPU bus interface
    - 2 MHz (when VCC = 5V)
  + 4-bit or 8-bit MPU interface enabled
  + 80 ´ 8-bit display RAM (80 characters max.)
  + 9,920-bit character generator ROM for a total of 240 character fonts
    - 208 character fonts (5 ´ 8 dot)
    - 32 character fonts (5 ´ 10 dot)
  + 64 ´ 8-bit character generator RAM
    - 8 character fonts (5 ´ 8 dot)
    - 4 character fonts (5 ´ 10 dot)
  + 16-common ´ 40-segment liquid crystal display driver
  + Programmable duty cycles
    - 1/8 for one line of 5 ´ 8 dots with cursor
    - 1/11 for one line of 5 ´ 10 dots with cursor
    - 1/16 for two lines of 5 ´ 8 dots with cursor
  + Wide range of instruction functions:
    - Display clear, cursor home, display on/off, cursor on/off, display character blink, cursor shift, display shift
  + Pin function compatibility with HD44780S
  + Automatic reset circuit that initializes the controller/driver after power on
  + Internal oscillator with external resistors
  + Low power consumption

2.4 Power Supply Unit

This supplies power to the entire circuit. Here we use a step down transformer to obtain the voltage from the mains. This unregulated voltage is regulated using full wave rectifiers and regulators, to obtain required voltage.

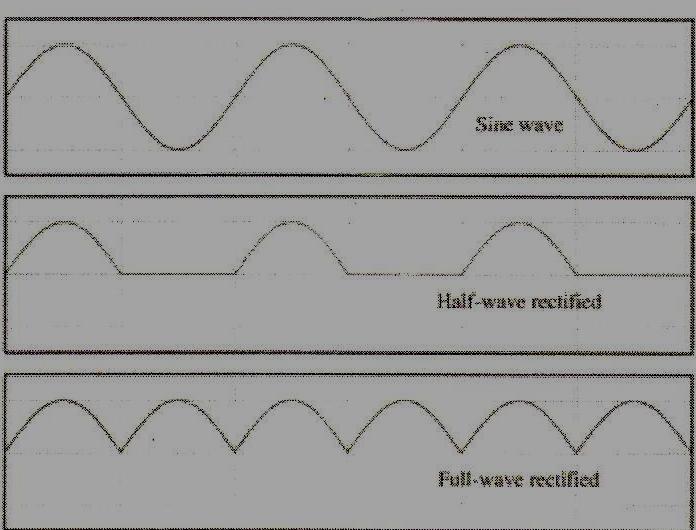


Figure 2.4.1:*Input and Output wave forms*

2.4.1 POWER SUPPLY CAPACITORS

This consists of two types of capacitors

1. Power supply filter capacitors
2. Power supply decoupling capacitors

2.4.2 Power supply filter capacitors

According to Thumbs rule for one milliamp current, one microfarad capacitor has to be used. Applying this rule we have used two capacitors

1. 470µF – This capacitor supplies current the microcontroller and LCD, which might require higher current, hence higher value of capacitance.
2. 100µF – This capacitor supplies current to the op-amp which might require lower current, hence lower value of capacitance.



Figure 2.4.2.1**:** *power supply capacitors*

2.4.3 Power supply decoupling capacitors

They are connected right across the supply pins. We know that energy can neither be created nor be destroyed. It can only be converted from one form to another. In the path of supplying power to the IC, there will be many wires or tracks which act as inductors. The property of inductor, according to Lenz’s law is that it does not allow sudden shoot up of current. Hence these capacitors act as storage devices which help in supplying current to ICs. Also all the circuits are operated only by a single supply because of which there might be disturbances in the supply. These capacitors do not allow disturbances in the power supply.

2.4.4 VOLTAGE REGULATOR

A voltage regulator is a circuit that supplies a constant voltage regardless of the changes in load currents. It may be used to produce a fixed DC output from variable DC (which contains small amount of AC in it). Although voltage regulators can be designed using op-amp, it is quicker and easier to use IC voltage regulators. IC voltage regulators are versatile and inexpensive and are available with features such as programmable output, current voltage boosting and internal short circuit current limiting, thermal shut down and floating operations for high voltage applications. IC voltage regulators are of following types;

* Fixed output voltage regulators
* Adjustable output voltage regulators
* Switching regulators
* Special regulators

Except switching regulator, all types of regulators are called linear regulators. Here we have used fixed output voltage regulators only in which positive voltage regulator is a type of it.

2.4.5 POSITIVE VOLTAGE REGULATOR

These ICs are designed as fixed voltage regulators and with adequate heat sinking can deliver output currents in excess of 1 A. The ICs have internal thermal overload protection and internal short circuit current limiting. Proper operation requires a common ground between input and output voltages. In addition to this, the difference between the input and output voltages called dropout voltage, must be typically 2.0 V.

Further a capacitance Ci is required if the voltage regulator is located at an appreciable distance from the power supply filter. Co is not needed but it may be required to improve the transient response of the regulator.

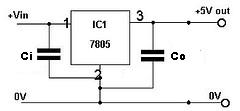
 

Figure2.4.5.1(a): *positive voltage regulator* Figure2.4.5.1(b): *7805 IC*

2.4.6 HEAT SINK

In the process of regulation, input voltage should be 3V more than the output voltage, in order to regulate. Due to this there will be a voltage across the regulator and current flowing through it. We know that P=VI, hence there will be generation of power which causes heating up of the components. Hence this heat has to be dissipated. This is done using **heat sink.**

A heat sink is an environment or object that absorbs and dissipates heat from another object using thermal contact (either direct or radiant). Heat sinks are used in a wide range of applications wherever efficient heat dissipation is required

Heat sinks function by efficiently transferring thermal energy ("heat") from an object at a relatively high temperature to a second object at a lower temperature with a much greater heat capacity. This rapid transfer of thermal energy quickly brings the first object into thermal equilibrium with the second, lowering the temperature of the first object, fulfilling the heat sink's role as a cooling device. Efficient function of a heat sink relies on rapid transfer of thermal energy from the first object to the heat sink, and the heat sink to the second object. The most common design of a heat sink is a metal device with many pins. The high thermal conductivity of the metal combined with its large surface area due to the pins results in the rapid transfer of thermal energy to the surrounding, cooler material. This cools the heat sink and whatever it is in direct thermal contact with. It is an anodized black component. It is black in color because black is a good radiator (black body radiation).

****

Figure 2.4.6.1**:** *Heat sink with aluminum holder to mount regulator*

* 1. Introduction to Serial Communication
     1. Introduction

The purpose of this application note is to attempt to describe the main elements in Serial Communication. This application note attempts to cover enough technical details of RS232, RS422 and RS485

* + 1. DCE and DTE Devices

DTE stands for Data Terminal Equipment, and DCE stands for Data Communications Equipment. These terms are used to indicate the pin-out for the connectors on a device and the direction of the signals on the pins. Your computer is a DTE device, while most other devices such as modem and other serial devices are usually DCE devices. RS-232 has been around as a standard for decades as an electrical interface between Data Terminal Equipment (DTE) and Data Circuit-Terminating Equipment (DCE) such as modems or DSUs. It appears under different incarnations such as RS-232C, RS-232D, V.24, V.28 or V.10. RS-232 is used for asynchronous data transfer as well as synchronous links such as SDLC, HDLC, Frame Relay and X.25

* + 1. Synchronous data transfer

In program-to-program communication, synchronous communication requires that each end of an exchange of communication respond in turn without initiating a new communication. A typical activity that might use a synchronous protocol would be a transmission of files from one point to another. As each transmission is received, a response is returned indicating success or the need to resend.

* + 1. Asynchronous data transfer

The term asynchronous is usually used to describe communications in which data can be transmitted intermittently rather than in a steady stream. For example, a telephone conversationis asynchronous because both parties can talk whenever they like. If the communication were synchronous, each party would be required to wait a specified interval before speaking. The difficulty with asynchronous communications is that the receiver must have a way to distinguish between valid data and noise. In computer communications, this is usually accomplished through a special start bit and stop bit at the beginning and end of each piece of data. For this reason, asynchronous communication is sometimes called start-stop transmission.

* + 1. RS232

RS-232 (Recommended standard-232) is a standard interface approved by the Electronic Industries Association (EIA) for connecting serial devices. In other words, RS-232 is a long established standard that describes the physical interface and protocol for relatively low-speed serial data communication between computers and related devices. An industry trade group, the Electronic Industries Association (EIA), defined it original for teletypewriter devices. In 1987, the EIA released a new version of the standard and change the name to EIA-232-D. Many people, however, still refer to the standard as RS 232C, or just RS-232. RS-232 is the interface that your computer uses to talk to and exchange data with your modem and other serial devices. The serial ports on most computers use a subset of the RS-232C standard.

* + 1. RS232 on DB9 (9-pin D-type connector)

There is a standardized pinout for RS-232 on a DB9 connector, as shown below

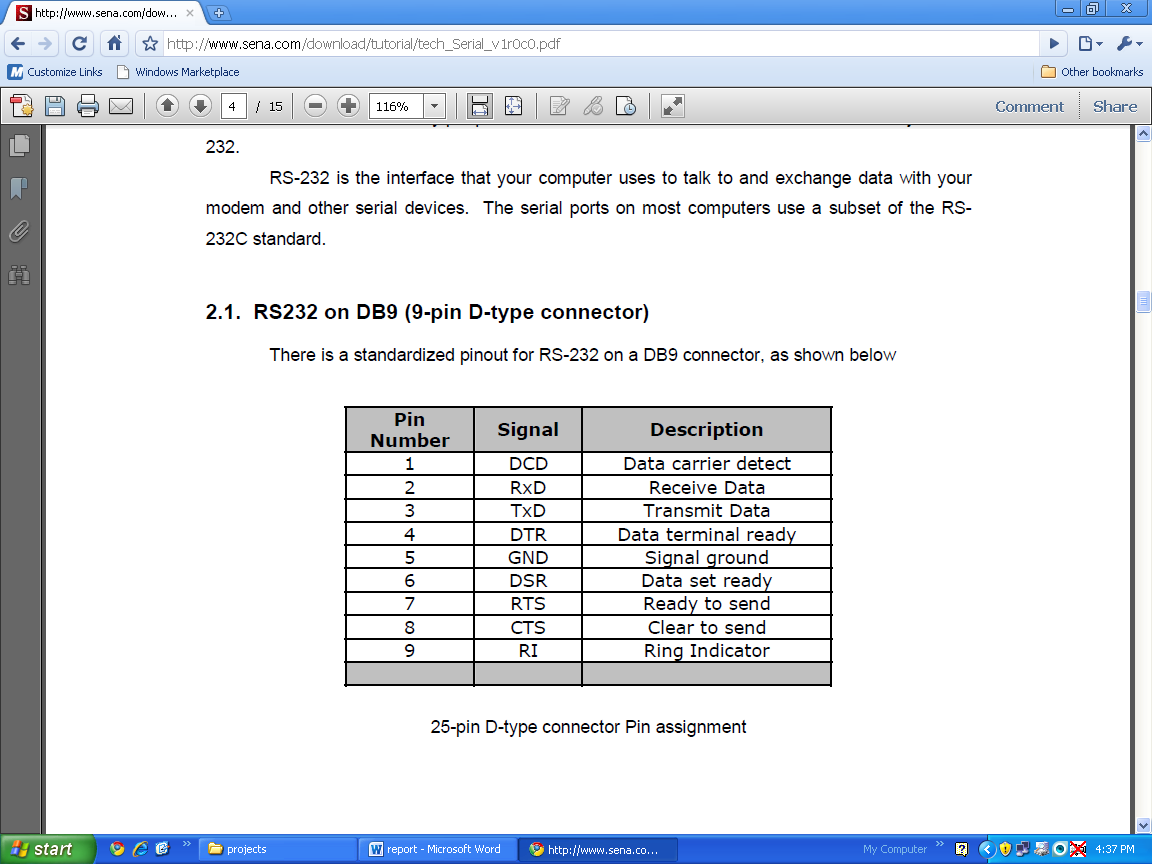


Table 2.5.6:pinout for RS-232

* + 1. RS232 on RJ-45

RJ-45 (Registered Jack-45) is an eight-wire connector used commonly to connect computers onto local-area networks (LAN), especially Ethernets. In other words, RJ-45 is a single-line jack for digital transmission over ordinary phone wire, either untwisted or twisted. The interface has eight pins or positions. For faster transmissions in which you're connecting to an Ethernet 10BASET network, you need to use twisted pair wire. RS232D, EIA/TIA - 561 standard is applied when connecting to or from a serial port with a 8 position Modular Jack (RJ45) though it is not widely used as such.



Figure 2.5.7.1: RJ-45 connector

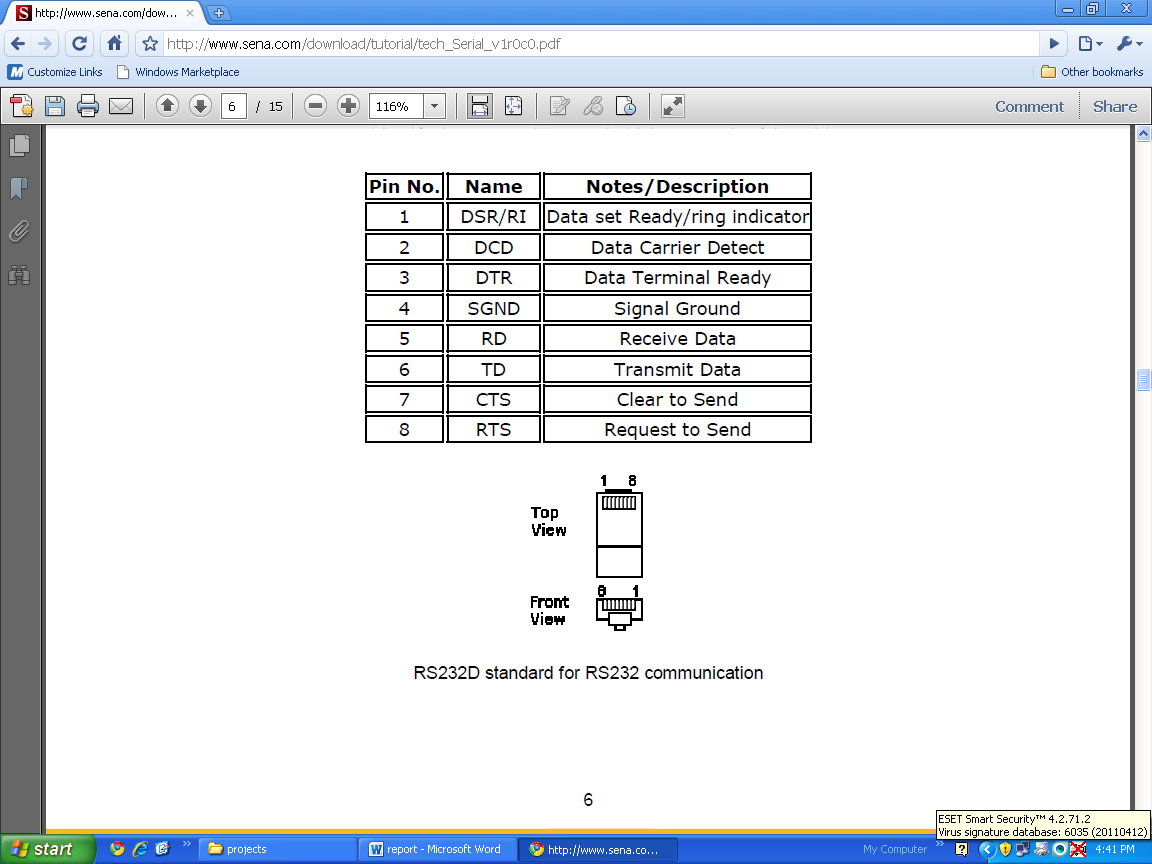


Table 2.5.7.2: Rs232 standard

* + 1. Signal Description
* TxD: - This pin carries data from the computer to the serial device
* RXD: - This pin carries data from the serial device to the computer
* DTR signals: - DTR is used by the computer to signal that it is ready to communicate with the serial device like modem. In other words, DTR indicates to the Dataset (i.e., the modem or DSU/CSU) that the DTE (computer) is ON.
* DSR: - Similarly to DTR, Data set ready (DSR) is an indication from the Dataset that it is ON.
* DCD: - Data Carrier Detect (DCD) indicates that carrier for the transmit data is ON.
* RTS: - This pin is used to request clearance to send data to a modem
* CTS: - This pin is used by the serial device to acknowledge the computer's RTS Signal. In most situations, RTS and CTS are constantly on throughout the communication session.
* Clock signals (TC, RC, and XTC): - The clock signals are only used for synchronouscommunications. The modem or DSU extracts the clock from the data stream and provides asteady clock signal to the DTE.
* Note that the transmit and receive clock signals do not have to be the same, or even at the same baud rate.
* CD: - CD stands for Carrier Detect. Carrier Detect is used by a modem to signal that it has amade a connection with another modem, or has detected a carrier tone. In other words, this issued by the modem to signal that a carrier signal has been received from a remote modem.
* RI: - RI stands for Ring Indicator. A modem toggles(keystroke) the state of this line when anincoming call rings your phone. In other words, this is used by an auto answer modem to signalthe receipt of a telephone ring signal
* The Carrier Detect (CD) and the Ring Indicator (RI) lines are only available in connections to amodem. Because most modems transmit status information to a PC when either a carrier signalis detected (i.e. when a connection is made to another modem) or when the line is ringing, thesetwo lines are rarely used.
  + 1. Limitations of RS-232

RS-232 has some serious shortcomings as an electrical interface.

Firstly, the interface presupposes a common ground between the DTE and DCE. This is a reasonable assumption where a short cable connects a DTE and DCE in the same room, but with longer lines and connections between devices that may be on different electrical busses, this may not be true. We have seen some spectacular electrical events causes by "uncommon grounds".

Secondly, a signal on a single line is impossible to screen effectively for noise. Byscreening the entire cable one can reduce the influence of outside noise, but internally generated noise remains a problem. As the baud rate and line length increase, the effect of capacitance between the cables introduces serious crosstalk until a point is reached where the data itself is unreadable.

Using low capacitance cable can reduce crosstalk. Also, as it is the higher frequencies that are the problem, control of slew rate in the signal (i.e., making the signal more rounded, rather than square) also decreases the crosstalk. The original specifications for RS-232 had no specification for maximum slew rate.

Voltage levels with respect to ground represent the RS 232 signals. There is a wire for each signal, together with the ground signal (reference for voltage levels). This interface is useful for point-to-point communication at slow speeds. For example, port COM1 in a PC can be used for a mouse, port COM2 for a modem, etc. This is an example of point-to-point communication: one port, one device. Due to the way the signals are connected, a common ground is required. This implies limited cable length - about 30 to 60 meters maximum. (Main problems are interference and resistance of the cable.) Shortly, RS 232 was designed for communication of local devices, and supports one transmitter and one receiver.

* 1. Interfacing a 4x4 Keyboard to an AT91 Microcontroller
     1. Introduction

This Application Note describes programming techniques implemented on the AT91 ARM-based microcontroller for scanning a 4x4 Keyboard matrix usually found in both consumer and industrial applications for numeric data entry.

* + 1. Keyboard interface

In this application, a 4x4 matrix keypad requiring eight Input/Output ports for interfacing is used as an example. Rows are connected to Peripheral Input/Output (PIO) pins configured as output. Columns are connected to PIO pins configured as input with interrupts. In this configuration, four pull-up resistors must be added in order to apply a high level on the corresponding input pins as shown in Figure 1. The corresponding hexadecimal value of the pressed key is sent on four LEDs.

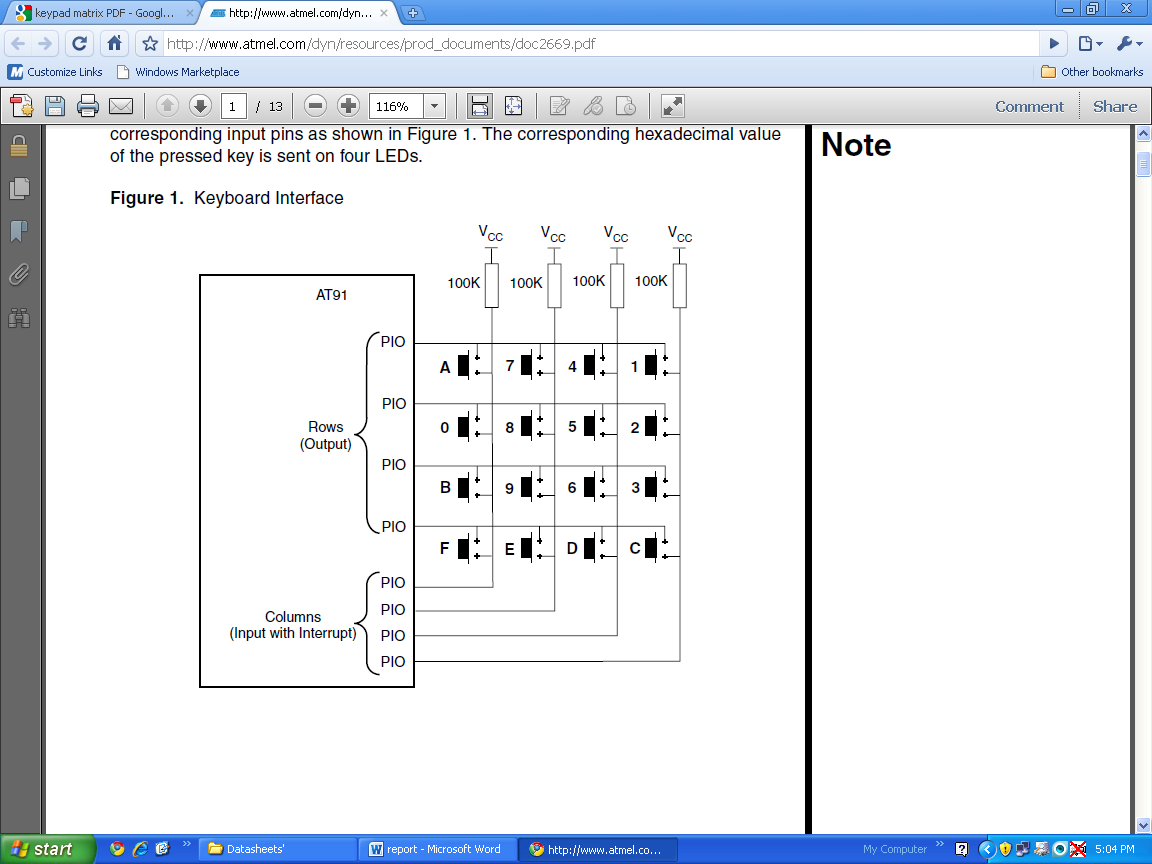


Figure 2.6.2.1: I/O configuration of keyboard

* + - 1. I/O configuration

Rows are connected to four PIO pins configured as outputs. Columns are connected to four PIO pins configured as inputs with interrupts. The idle state of these pins is high level due to four pull-up resistors. PIO interrupt is generated by a low level applied to these pins (caused by a key pressed). Four additional PIO pins are configured as outputs to send the value of the pressed key to LEDS.

* + - 1. Timer Counter Configuration

The Timer Counter is configured in waveform operating mode with RC compare interrupt. The Timer Counter is initialized to be incremented on internal clock cycles. The debouncing time is programmable by initializing the RC compare register value according to the clock source selected. A software trigger is used to reset the timer counter and start the counter clock.

* + - 1. Interrupt

When a key is pressed, a low level is applied to the pin corresponding to the column associated to the key (pins configured as inputs with interrupts). A falling edge applied to a column pin creates a PIO interrupt. Then, the processor executes the PIO interrupt subroutine (debouncing) and comes back to its previous state (in the main program). After debouncing time, a RC compare timer interrupt occurs and the processor then executes the timer interrupt subroutine (decoding the pressed key) and comes back to its previous state (in the main program).

* + - 1. Keyboard Scan

The Keyboard used is a 4x4 matrixes Keyboard. Columns are connected to pins configured as inputsand having the input change interrupt enabled. The initial state of these pins is high level due to four external pull-up resistors. The state machine is initialized to start with fast scan which outputs zeroes to all rowsand detects all keys at the same time. When a key is pressed, a low level is applied to the corresponding column and causes a PIO interrupt to detect the first edge. Once any key is detected, debouncing is started. The attempt to press a key on a physical keypad and have this activity detected can fail as a result of several noise sources, glitches, spikes, etc., to mention some of the possible causes of de bounce problems. The timer is used to eliminate all noise of less than a few milliseconds. Normally this is dependent on the mechanical characteristics of the keys. In this application example, a 20ms programmable de bouncing time is used. After debouncing is completed, a detailed scan is executed. A second fast scan is done to assure that any detection made during the first fast scan stage was not just noise. Then, rows are configured as inputs. When a key is pressed a high level is applied in the corresponding row.

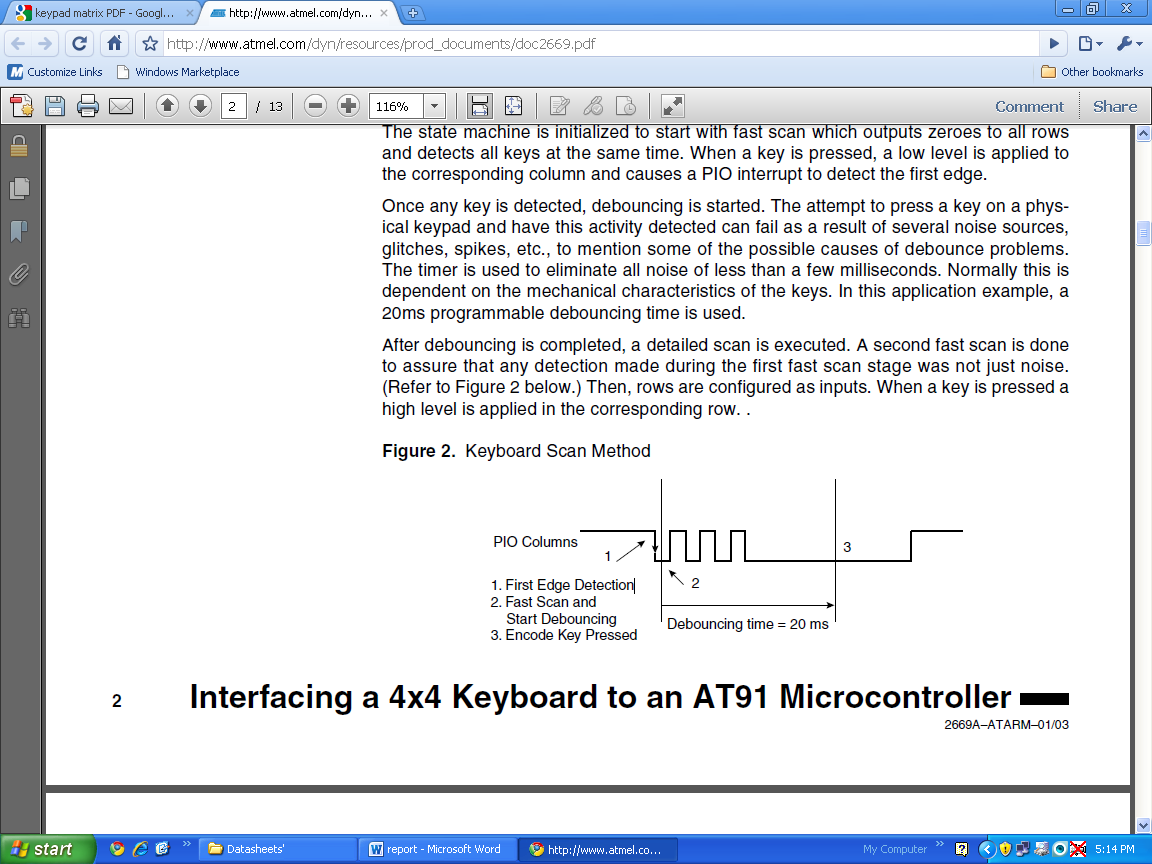


Figure 2.6.2.4: *Keyboard Scan Method*

**3. Functional description**

3.1 Architectural overview

The ARM7TDMI-S is a general purpose 32-bit microprocessor, which offers high

performance and very low power consumption. The ARM architecture is based on

Reduced Instruction Set Computer (RISC) principles, and the instruction set and related

decode mechanism are much simpler than those of microprogrammed Complex

Instruction Set Computers (CISC). This simplicity results in a high instruction throughput

and impressive real-time interrupt response from a small and cost-effective processor

core.

Pipeline techniques are employed so that all parts of the processing and memory systems

can operate continuously. Typically, while one instruction is being executed, its successor

is being decoded, and a third instruction is being fetched from memory.

The ARM7TDMI-S processor also employs a unique architectural strategy known as

Thumb, which makes it ideally suited to high-volume applications with memory

restrictions, or applications where code density is an issue.

The key idea behind Thumb is that of a super-reduced instruction set. Essentially, the

ARM7TDMI-S processor has two instruction sets:

**•** The standard 32-bit ARM set.

**•** A 16-bit Thumb set.

The Thumb set’s 16-bit instruction length allows it to approach twice the density of

standard ARM code while retaining most of the ARM’s performance advantage over a

traditional 16-bit processor using 16-bit registers. This is possible because Thumb code

operates on the same 32-bit register set as ARM code.

Thumb code is able to provide up to 65 % of the code size of ARM, and 160 % of the

performance of an equivalent ARM processor connected to a 16-bit memory system.

The particular flash implementation in the LPC2141/42/44/46/48 allows for full speed

execution also in ARM mode. It is recommended to program performance critical and

short code sections (such as interrupt service routines and DSP algorithms) in ARM

mode. The impact on the overall code size will be minimal but the speed can be increased

by 30 % over Thumb mode.

**3.2 On-chip flash program memory**

The LPC2141/42/44/46/48 incorporate a 32 kB, 64 kB, 128 kB, 256 kB and 512 kB flash

memory system respectively. This memory may be used for both code and data storage.

Programming of the flash memory may be accomplished in several ways. It may be

programmed In System via the serial port. The application program may also erase and/or

program the flash while the application is running, allowing a great degree of flexibility for

data storage field firmware upgrades, etc. Due to the architectural solution chosen for an

on-chip boot loader, flash memory available for user’s code on LPC2141/42/44/46/48 is

32 kB

**3.3 On-chip static RAM**

On-chip static RAM may be used for code and/or data storage. The SRAM may be

accessed as 8-bit, 16-bit, and 32-bit. The LPC2141, LPC2142/44 and LPC2146/48

provide 8 kB, 16 kB and 32 kB of static RAM respectively.

In case of LPC2146/48 only, an 8 kB SRAM block intended to be utilized mainly by the

USB can also be used as a general purpose RAM

**3.4 Interrupt controller**

The Vectored Interrupt Controller (VIC) accepts all of the interrupt request inputs and

categorizes them as Fast Interrupt Request (FIQ), vectored Interrupt Request (IRQ), and

non-vectored IRQ as defined by programmable settings. The programmable assignment

scheme means that priorities of interrupts from the various peripherals can be dynamically

assigned and adjusted.

Fast interrupt request (FIQ) has the highest priority. If more than one request is assigned

to FIQ, the VIC combines the requests to produce the FIQ signal to the ARM processor.

The fastest possible FIQ latency is achieved when only one request is classified as FIQ,

because then the FIQ service routine does not need to branch into the interrupt service

routine but can run from the interrupt vector location. If more than one request is assigned

to the FIQ class, the FIQ service routine will read a word from the VIC that identifies which

FIQ source(s) is (are) requesting an interrupt.

Vectored IRQs have the middle priority. Sixteen of the interrupt requests can be assigned

to this category. Any of the interrupt requests can be assigned to any of the 16 vectored

IRQ slots, among which slot 0 has the highest priority and slot 15 has the lowest.

Non-vectored IRQs have the lowest priority.

The VIC combines the requests from all the vectored and non-vectored IRQs to produce

the IRQ signal to the ARM processor. The IRQ service routine can start by reading a

register from the VIC and jumping there. If any of the vectored IRQs are pending, the VIC

provides the address of the highest-priority requesting IRQs service routine, otherwise it

provides the address of a default routine that is shared by all the non-vectored IRQs. The

default routine can read another VIC register to see what IRQs are active.

**3.4.1 Interrupt sources**

Each peripheral device has one interrupt line connected to the Vectored Interrupt

Controller, but may have several internal interrupt flags. Individual interrupt flags may also

represent more than one interrupt source.

**3.5 Pin connect block**

The pin connect block allows selected pins of the microcontroller to have more than one

function. Configuration registers control the multiplexers to allow connection between the

pin and the on chip peripherals. Peripherals should be connected to the appropriate pins

prior to being activated, and prior to any related interrupt(s) being enabled. Activity of any

enabled peripheral function that is not mapped to a related pin should be considered

undefined.

The Pin Control Module with its pin select registers defines the functionality of the

microcontroller in a given hardware environment.

After reset all pins of Port 0 and Port 1 are configured as input with the following

exceptions: If debug is enabled, the JTAG pins will assume their JTAG functionality; if

trace is enabled, the Trace pins will assume their trace functionality. The pins associated

with the I2C0 and I2C1 interface are open drain.

**3.6 Fast general purpose parallel I/O**

Device pins that are not connected to a specific peripheral function are controlled by the

GPIO registers. Pins may be dynamically configured as inputs or outputs. Separate

registers allow setting or clearing any number of outputs simultaneously. The value of the

output register may be read back, as well as the current state of the port pins.

LPC2141/42/44/46/48 introduce accelerated GPIO functions over prior LPC2000 devices:

**•** GPIO registers are relocated to the ARM local bus for the fastest possible I/O timing.

**•** Mask registers allow treating sets of port bits as a group, leaving other bits

unchanged.

**•** All GPIO registers are byte addressable.

**•** Entire port value can be written in one instruction.

**3.6.1 Features**

**•** Bit-level set and clear registers allow a single instruction set or clear of any number of

bits in one port.

**•** Direction control of individual bits.

**•** Separate control of output set and clear.

**•** All I/O default to inputs after reset.

**3.7 10-bit ADC**

The LPC2141/42 contain one and the LPC2144/46/48 contain two analog to digital

converters. These converters are single 10-bit successive approximation analog to digital

converters. While ADC0 has six channels, ADC1 has eight channels. Therefore, total

number of available ADC inputs for LPC2141/42 is 6 and for LPC2144/46/48 is 14.

**3.7.1 Features**

**•** 10 bit successive approximation analog to digital converter.

**•** Measurement range of 0 V to VREF (2.0 V VREF VDDA).

**•** Each converter capable of performing more than 400,000 10-bit samples per second.

**•** Every analog input has a dedicated result register to reduce interrupt overhead.

**•** Burst conversion mode for single or multiple inputs.

**•** Optional conversion on transition on input pin or timer match signal.

**•** Global Start command for both converters (LPC2142/44/46/48 only).

**3.8 10-bit DAC**

The DAC enables the LPC2141/42/44/46/48 to generate a variable analog output. The

maximum DAC output voltage is the VREF voltage.

**3.8.1 Features**

**•** 10-bit DAC.

**•** Buffered output.

**•** Power-down mode available.

**3.9 USB 2.0 device controller**

The USB is a 4-wire serial bus that supports communication between a host and a

number (127 max) of peripherals. The host controller allocates the USB bandwidth to

attached devices through a token based protocol. The bus supports hot plugging,

unplugging, and dynamic configuration of the devices. All transactions are initiated by the

host controller.

The LPC2141/42/44/46/48 is equipped with a USB device controller that enables

12 Mbit/s data exchange with a USB host controller. It consists of a register interface,

serial interface engine, endpoint buffer memory and DMA controller. The serial interface

engine decodes the USB data stream and writes data to the appropriate end point buffer

memory. The status of a completed USB transfer or error condition is indicated via status

registers. An interrupt is also generated if enabled.

A DMA controller (available in LPC2146/48 only) can transfer data between an endpoint

buffer and the USB RAM.

**3.9.1 Features**

**•** Fully compliant with USB 2.0 Full-speed specification.

**•** Supports 32 physical (16 logical) endpoints.

**•** Supports control, bulk, interrupt and isochronous endpoints.

**•** Scalable realization of endpoints at run time.

**•** Endpoint maximum packet size selection (up to USB maximum specification) by

software at run time.

**•** RAM message buffer size based on endpoint realization and maximum packet size.

**•** Supports SoftConnect and GoodLink LED indicator. These two functions are sharing

one pin.

**•** Supports bus-powered capability with low suspend current.

**•** Supports DMA transfer on all non-control endpoints (LPC2146/48 only).

**•** One duplex DMA channel serves all endpoints (LPC2146/48 only).

**•** Allows dynamic switching between CPU controlled and DMA modes (only in

LPC2146/48).

**•** Double buffer implementation for bulk and isochronous endpoints.

**3.10 UARTs**

The LPC2141/42/44/46/48 each contain two UARTs. In addition to standard transmit and

receive data lines, the LPC2144/46/48 UART1 also provides a full modem control

handshake interface.

Compared to previous LPC2000 microcontrollers, UARTs in LPC2141/42/44/46/48

introduce a fractional baud rate generator for both UARTs, enabling these microcontrollers

to achieve standard baud rates such as 115200 with any crystal frequency above 2 MHz.

In addition, auto-CTS/RTS flow-control functions are fully implemented in hardware

(UART1 in LPC2144/46/48 only).

**3.10.1 Features**

**•** 16 byte Receive and Transmit FIFOs.

**•** Register locations conform to ‘550 industry standard.

**•** Receiver FIFO trigger points at 1, 4, 8, and 14 bytes

**•** Built-in fractional baud rate generator covering wide range of baud rates without a

need for external crystals of particular values.

**•** Transmission FIFO control enables implementation of software (XON/XOFF) flow

control on both UARTs.

**•** LPC2144/46/48 UART1 equipped with standard modem interface signals. This

module also provides full support for hardware flow control (auto-CTS/RTS).

**3.11 I2C-bus serial I/O controller**

The LPC2141/42/44/46/48 each contain two I2C-bus controllers.

The I2C-bus is bidirectional, for inter-IC control using only two wires: a serial clock line

(SCL), and a serial data line (SDA). Each device is recognized by a unique address and

can operate as either a receiver-only device (e.g., an LCD driver or a transmitter with the

capability to both receive and send information (such as memory)). Transmitters and/or

receivers can operate in either master or slave mode, depending on whether the chip has

to initiate a data transfer or is only addressed. The I2C-bus is a multi-master bus, it can be

controlled by more than one bus master connected to it.

The I2C-bus implemented in LPC2141/42/44/46/48 supports bit rates up to 400 kbit/s

(Fast I2C-bus).

**3.11.1 Features**

**•** Compliant with standard I2C-bus interface.

**•** Easy to configure as master, slave, or master/slave.

**•** Programmable clocks allow versatile rate control.

**•** Bidirectional data transfer between masters and slaves.

**•** Multi-master bus (no central master).

**•** Arbitration between simultaneously transmitting masters without corruption of serial

data on the bus.

**•** Serial clock synchronization allows devices with different bit rates to communicate via

one serial bus.

**•** Serial clock synchronization can be used as a handshake mechanism to suspend and

resume serial transfer.

**•** The I2C-bus can be used for test and diagnostic purposes.

**3.12 SPI serial I/O controller**

The LPC2141/42/44/46/48 each contain one SPI controller. The SPI is a full duplex serial

interface, designed to handle multiple masters and slaves connected to a given bus. Only

a single master and a single slave can communicate on the interface during a given data

transfer. During a data transfer the master always sends a byte of data to the slave, and

the slave always sends a byte of data to the master.

**3.12.1 Features**

**•** Compliant with SPI specification.

**•** Synchronous, Serial, Full Duplex, Communication.

**•** Combined SPI master and slave.

**•** Maximum data bit rate of one eighth of the input clock rate.

**3.13 SSP serial I/O controller**

The LPC2141/42/44/46/48 each contain one SSP. The SSP controller is capable of

operation on a SPI, 4-wire SSI, or Microwire bus. It can interact with multiple masters and

slaves on the bus. However, only a single master and a single slave can communicate on

the bus during a given data transfer. The SSP supports full duplex transfers, with data

frames of 4 bits to 16 bits of data flowing from the master to the slave and from the slave

to the master. Often only one of these data flows carries meaningful data.

**3.13.1 Features**

**•** Compatible with Motorola’s SPI, TI’s 4-wire SSI and National Semiconductor’s

Microwire buses.

**•** Synchronous serial communication.

**•** Master or slave operation.

**•** 8-frame FIFOs for both transmit and receive.

**•** Four bits to 16 bits per frame.

**3.14 General purpose timers/external event counters**

The Timer/Counter is designed to count cycles of the peripheral clock (PCLK) or an

externally supplied clock and optionally generate interrupts or perform other actions at

specified timer values, based on four match registers. It also includes four capture inputs

to trap the timer value when an input signal transitions, optionally generating an interrupt.

Multiple pins can be selected to perform a single capture or match function, providing an

application with ‘or’ and ‘and’, as well as ‘broadcast’ functions among them.

The LPC2141/42/44/46/48 can count external events on one of the capture inputs if the

minimum external pulse is equal or longer than a period of the PCLK. In this configuration,

unused capture lines can be selected as regular timer capture inputs, or used as external

interrupts.

**3.14.1 Features**

**•** A 32-bit timer/counter with a programmable 32-bit prescaler.

**•** External event counter or timer operation.

**•** Four 32-bit capture channels per timer/counter that can take a snapshot of the timer

value when an input signal transitions. A capture event may also optionally generate

an interrupt.

**•** Four 32-bit match registers that allow:

**–** Continuous operation with optional interrupt generation on match.

**–** Stop timer on match with optional interrupt generation.

**–** Reset timer on match with optional interrupt generation.

**•** Four external outputs per timer/counter corresponding to match registers, with the

following capabilities:

**–** Set LOW on match.

**–** Set HIGH on match.

**–** Toggle on match.

**–** Do nothing on match.

**3.15 Watchdog timer**

The purpose of the watchdog is to reset the microcontroller within a reasonable amount of

time if it enters an erroneous state. When enabled, the watchdog will generate a system

reset if the user program fails to ‘feed’ (or reload) the watchdog within a predetermined

amount of time.

**3.15.1 Features**

**•** Internally resets chip if not periodically reloaded.

**•** Debug mode.

**•** Enabled by software but requires a hardware reset or a watchdog reset/interrupt to be

disabled.

**•** Incorrect/Incomplete feed sequence causes reset/interrupt if enabled.

**•** Flag to indicate watchdog reset.

**•** Programmable 32-bit timer with internal pre-scaler.

**•** Selectable time period from (TPCLK 256 4) to (TPCLK 232 4) in multiples of

TPCLK 4.

**3.16 Real-time clock**

The RTC is designed to provide a set of counters to measure time when normal or idle

operating mode is selected. The RTC has been designed to use little power, making it

suitable for battery powered systems where the CPU is not running continuously (Idle

mode).

**3.16.1 Features**

**•** Measures the passage of time to maintain a calendar and clock.

**•** Ultra-low power design to support battery powered systems.

**•** Provides Seconds, Minutes, Hours, Day of Month, Month, Year, Day ofWeek, and Day

of Year.

**•** Can use either the RTC dedicated 32 kHz oscillator input or clock derived from the

external crystal/oscillator input at XTAL1. Programmable reference clock divider

allows fine adjustment of the RTC.

**•** Dedicated power supply pin can be connected to a battery or the main 3.3 V.

**3.17 Pulse width modulator**

The PWM is based on the standard timer block and inherits all of its features, although

only the PWM function is pinned out on the LPC2141/42/44/46/48. The timer is designed

to count cycles of the peripheral clock (PCLK) and optionally generate interrupts or

perform other actions when specified timer values occur, based on seven match registers.

The PWM function is also based on match register events.

The ability to separately control rising and falling edge locations allows the PWM to be

used for more applications. For instance, multi-phase motor control typically requires three

non-overlapping PWM outputs with individual control of all three pulse widths and

positions.

Two match registers can be used to provide a single edge controlled PWM output. One

match register (MR0) controls the PWM cycle rate, by resetting the count upon match.

The other match register controls the PWM edge position. Additional single edge

controlled PWM outputs require only one match register each, since the repetition rate is

the same for all PWM outputs. Multiple single edge controlled PWM outputs will all have a

rising edge at the beginning of each PWM cycle, when an MR0 match occurs.

Three match registers can be used to provide a PWM output with both edges controlled.

Again, the MR0 match register controls the PWM cycle rate. The other match registers

control the two PWM edge positions. Additional double edge controlled PWM outputs

require only two match registers each, since the repetition rate is the same for all PWM

outputs.

With double edge controlled PWM outputs, specific match registers control the rising and

falling edge of the output. This allows both positive going PWM pulses (when the rising

edge occurs prior to the falling edge), and negative going PWM pulses (when the falling

edge occurs prior to the rising edge).

**3.17.1 Features**

**•** Seven match registers allow up to six single edge controlled or three double edge

controlled PWM outputs, or a mix of both types.

**•** The match registers also allow:

**–** Continuous operation with optional interrupt generation on match.

**–** Stop timer on match with optional interrupt generation.

**–** Reset timer on match with optional interrupt generation.

**•** Supports single edge controlled and/or double edge controlled PWM outputs. Single

edge controlled PWM outputs all go HIGH at the beginning of each cycle unless the

output is a constant LOW. Double edge controlled PWM outputs can have either edge

occur at any position within a cycle. This allows for both positive going and negative

going pulses.

**•** Pulse period and width can be any number of timer counts. This allows complete

flexibility in the trade-off between resolution and repetition rate. All PWM outputs will

occur at the same repetition rate.

**•** Double edge controlled PWM

**•** Match register updates are synchronized with pulse outputs to prevent generation of

erroneous pulses. Software must ‘release’ new match values before they can become

effective.

**•** May be used as a standard timer if the PWM mode is not enabled.

**•** A 32-bit Timer/Counter with a programmable 32-bit Pre scalar.

**3.18 System control**

**3.18.1 Crystal oscillator**

On-chip integrated oscillator operates with external crystal in range of 1 MHz to 25 MHz.

The oscillator output frequency is called fosc and the ARM processor clock frequency is

referred to as CCLK for purposes of rate equations, etc. fosc and CCLK are the same value

unless the PLL is running and connected.

**3.18.2 PLL**

The PLL accepts an input clock frequency in the range of 10 MHz to 25 MHz. The input

frequency is multiplied up into the range of 10 MHz to 60 MHz with a Current Controlled

Oscillator (CCO). The multiplier can be an integer value from 1 to 32 (in practice, the

multiplier value cannot be higher than 6 on this family of microcontrollers due to the upper

frequency limit of the CPU). The CCO operates in the range of 156 MHz to 320 MHz, so

there is an additional divider in the loop to keep the CCO within its frequency range while

the PLL is providing the desired output frequency. The output divider may be set to divide

by 2, 4, 8, or 16 to produce the output clock. Since the minimum output divider value is 2,

it is insured that the PLL output has a 50 % duty cycle. The PLL is turned off and

bypassed following a chip reset and may be enabled by software. The program must

configure and activate the PLL, wait for the PLL to Lock, then connect to the PLL as a

clock source. The PLL settling time is 100 s.

**3.18.3 Reset and wake-up timer**

Reset has two sources on the LPC2141/42/44/46/48: the RESET pin and watchdog reset.

The RESET pin is a Schmitt trigger input pin with an additional glitch filter. Assertion of

chip reset by any source starts the Wake-up Timer (see Wake-up Timer description

below), causing the internal chip reset to remain asserted until the external reset is

de-asserted, the oscillator is running, a fixed number of clocks have passed, and the

on-chip flash controller has completed its initialization.

When the internal reset is removed, the processor begins executing at address 0, which is

the reset vector. At that point, all of the processor and peripheral registers have been

initialized to predetermined values.

The Wake-up Timer ensures that the oscillator and other analog functions required for

chip operation are fully functional before the processor is allowed to execute instructions.

This is important at power on, all types of reset, and whenever any of the aforementioned

functions are turned off for any reason. Since the oscillator and other functions are turned

off during Power-down mode, any wake-up of the processor from Power-down mode

makes use of the Wake-up Timer.

The Wake-up Timer monitors the crystal oscillator as the means of checking whether it is

safe to begin code execution. When power is applied to the chip, or some event caused

the chip to exit Power-down mode, some time is required for the oscillator to produce a

signal of sufficient amplitude to drive the clock logic. The amount of time depends on

many factors, including the rate of VDD ramp (in the case of power on), the type of crystal

and its electrical characteristics (if a quartz crystal is used), as well as any other external

circuitry (e.g. capacitors), and the characteristics of the oscillator itself under the existing

ambient conditions.

**3.18.4 Brownout detector**

The LPC2141/42/44/46/48 include 2-stage monitoring of the voltage on the VDD pins. If

this voltage falls below 2.9 V, the BOD asserts an interrupt signal to the VIC. This signal

can be enabled for interrupt; if not, software can monitor the signal by reading dedicated

register.

The second stage of low voltage detection asserts reset to inactivate the

LPC2141/42/44/46/48 when the voltage on the VDD pins falls below 2.6 V. This reset

prevents alteration of the flash as operation of the various elements of the chip would

otherwise become unreliable due to low voltage. The BOD circuit maintains this reset

down below 1 V, at which point the POR circuitry maintains the overall reset.

Both the 2.9 V and 2.6 V thresholds include some hysteresis. In normal operation, this

hysteresis allows the 2.9 V detection to reliably interrupt, or a regularly-executed event

loop to sense the condition.

**3.18.5 Code security**

This feature of the LPC2141/42/44/46/48 allow an application to control whether it can be

debugged or protected from observation.

If after reset on-chip boot loader detects a valid checksum in flash and reads 0x8765 4321

from address 0x1FC in flash, debugging will be disabled and thus the code in flash will be

protected from observation. Once debugging is disabled, it can be enabled only by

performing a full chip erase using the ISP.

**3.18.6 External interrupt inputs**

The LPC2141/42/44/46/48 include up to nine edge or level sensitive External Interrupt

Inputs as selectable pin functions. When the pins are combined, external events can be

processed as four independent interrupt signals. The External Interrupt Inputs can

optionally be used to wake-up the processor from Power-down mode.

Additionally capture input pins can also be used as external interrupts without the option

to wake the device up from Power-down mode.

**3.18.7 Memory mapping control**

The Memory Mapping Control alters the mapping of the interrupt vectors that appear

beginning at address 0x0000 0000. Vectors may be mapped to the bottom of the on-chip

flash memory, or to the on-chip static RAM.

**3.18.8 Power control**

The LPC2141/42/44/46/48 supports two reduced power modes: Idle mode and

Power-down mode.

In Idle mode, execution of instructions is suspended until either a reset or interrupt occurs.

Peripheral functions continue operation during Idle mode and may generate interrupts to

cause the processor to resume execution. Idle mode eliminates power used by the

processor itself, memory systems and related controllers, and internal buses.

In Power-down mode, the oscillator is shut down and the chip receives no internal clocks.

The processor state and registers, peripheral registers, and internal SRAM values are

preserved throughout Power-down mode and the logic levels of chip output pins remain

static. The Power-down mode can be terminated and normal operation resumed by either

a reset or certain specific interrupts that are able to function without clocks. Since all

dynamic operation of the chip is suspended, Power-down mode reduces chip power

consumption to nearly zero.

Selecting an external 32 kHz clock instead of the PCLK as a clock-source for the on-chip

RTC will enable the microcontroller to have the RTC active during Power-down mode.

Power-down current is increased with RTC active. However, it is significantly lower than in

Idle mode.

A Power Control for Peripherals feature allows individual peripherals to be turned off if

they are not needed in the application, resulting in additional power savings during active

and Idle mode.

**3.18.9 VPB bus**

The VPB divider determines the relationship between the processor clock (CCLK) and the

clock used by peripheral devices (PCLK). The VPB divider serves two purposes. The first

is to provide peripherals with the desired PCLK via VPB bus so that they can operate at

the speed chosen for the ARM processor. In order to achieve this, the VPB bus may be

slowed down to 12 to 14 of the processor clock rate. Because the VPB bus must work

properly at power-up (and its timing cannot be altered if it does not work since the VPB

divider control registers reside on the VPB bus), the default condition at reset is for the

VPB bus to run at 14 of the processor clock rate. The second purpose of the VPB divider

is to allow power savings when an application does not require any peripherals to run at

the full processor rate. Because the VPB divider is connected to the PLL output, the PLL

remains active (if it was running) during Idle mode.

**3.19 Emulation and debugging**

The LPC2141/42/44/46/48 support emulation and debugging via a JTAG serial port. A

trace port allows tracing program execution. Debugging and trace functions are

multiplexed only with GPIOs on Port 1. This means that all communication, timer and

interface peripherals residing on Port 0 are available during the development and

debugging phase as they are when the application is run in the embedded system itself.

**3.19.1 EmbeddedICE**

Standard ARM EmbeddedICE logic provides on-chip debug support. The debugging of

the target system requires a host computer running the debugger software and an

EmbeddedICE protocol convertor.EmbeddedICE protocol convertor converts the remote

debug protocol commands to the JTAG data needed to access the ARM core.

The ARM core has a Debug Communication Channel (DCC) function built-in. The DCC

allows a program running on the target to communicate with the host debugger or another

separate host without stopping the program flow or even entering the debug state. The

DCC is accessed as a co-processor 14 by the program running on the ARM7TDMI-S

core. The DCC allows the JTAG port to be used for sending and receiving data without

affecting the normal program flow. The DCC data and control registers are mapped in to

addresses in the EmbeddedICE logic.

**3.19.2 Embedded trace**

Since the LPC2141/42/44/46/48 have significant amounts of on-chip memory, it is not

possible to determine how the processor core is operating simply by observing the

external pins. The Embedded Trace Macrocell (ETM) provides real-time trace capability

for deeply embedded processor cores. It outputs information about processor execution to

the trace port.

The ETM is connected directly to the ARM core and not to the main AMBA system bus. It

compresses the trace information and exports it through a narrow trace port. An external

trace port analyzer must capture the trace information under software debugger control.

Instruction trace (or PC trace) shows the flow of execution of the processor and provides a

list of all the instructions that were executed. Instruction trace is significantly compressed

by only broadcasting branch addresses as well as a set of status signals that indicate the

pipeline status on a cycle by cycle basis. Trace information generation can be controlled

by selecting the trigger resource. Trigger resources include address comparators,

counters and sequencers. Since trace information is compressed the software debugger

requires a static image of the code being executed. Self-modifying code can not be traced

because of this restriction.

**3.19.3 RealMonitor**

RealMonitor is a configurable software module, developed by ARM Inc., which enables

real-time debug. It is a lightweight debug monitor that runs in the background while users

debug their foreground application. It communicates with the host using the DCC, which is

present in the EmbeddedICE logic. The LPC2141/42/44/46/48 contain a specific

configuration of RealMonitor software programmed into the on-chip flash memory.

4.  RF  Module  Operation

4.1. Serial Communications

The XBee/XBee-PRO OEM RF Modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device (For example: Through a Max- Stream proprietary RS-232 or USB interface board).

**4.1.1. UART Data Flow**



Figure 4.1.1.1(a): System  Data  Flow  Diagram  in  a  UART‐interfaced  environmentLow‐asserted  signals  distinguished  with  horizontal  line  over  signal  name.

**Serial Data**

Data enters the module UART through the DI pin (pin 3) as an asynchronous serial signal. The sig-nal should idle high when no data is being transmitted. Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). The following figure illustrates the serial bit pattern of data passing through the module.



Figure 4.1.1.1(b): UART  data  packet  0x1F  (decimal  number  ʺ31ʺ)  as  transmitted

through  the  RF  moduleExample  Data  Format  is  8‐N‐1  (bits  ‐ parity  ‐ #  of  stop  bits

The module UART performs tasks, such as timing and parity checking, that are needed for data communications. Serial communications depend on the two UARTs to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits).

**4.1.2. Transparent Operation**

**Serial-to-RF Packetization**By default, XBee/XBee-PRO RF Modules operate in Transparent Mode. When operating in this mode, the modules act as a serial line replacement - all UART data received through the DI pin is queued up for RF transmission. When RF data is received, the data is sent out the DO pin.

Data is buffered in the DI buffer until one of the following causes the data to be packetized and transmitted:

1. No serial characters are received for the amount of time determined by the RO (Packetiza-tion Timeout) parameter. If RO = 0, packetization begins when a character is received. 2. The maximum number of characters that will fit in an RF packet (100) is received. 3. The Command Mode Sequence (GT + CC + GT) is received. Any character buffered in the DI buffer before the sequence is transmitted.

If the module cannot immediately transmit (for instance, if it is already receiving RF data), the serial data is stored in the DI Buffer. The data is packetized and sent at any RO timeout or when 100 bytes (maximum packet size) are received. If the DI buffer becomes full, hardware or software flow control must be implemented in order to prevent overflow (loss of data between the host and module).

**4.1.3. API Operation**

API (Application Programming Interface) Operation is an alternative to the default Transparent Operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module. When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the module. Transmit Data Frames (received through the DI pin (pin 3)) include: • RF Transmit Data Frame • Command Frame (equivalent to AT commands) Receive Data Frames (sent out the DO pin (pin 2)) include: • RF-received data frame • Command response • Event notifications such as reset, associate, disassociate, etc. The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the module that contain address and payload information instead of using command mode to modify addresses. The module will send data frames to the application containing status packets; as well as source, RSSI and payload information from received data packets. The API operation option facilitates many operations such as the examples cited below:

-> Transmitting data to multiple destinations without entering Command Mode -> Receive success/failure status of each transmitted RF packet -> Identify the source address of each received packet

**4.1.4. Flow Control**



Figure  4.1.4.1: Internal  Data  Flow  Diagram

**DI (Data In) Buffer**

When serial data enters the RF module through the DI pin (pin 3), the data is stored in the DI Buffer until it can be processed. **Hardware Flow Control (CTS).** When the DI buffer is 17 bytes away from being full; by default, the module de-asserts CTS (high) to signal to the host device to stop sending data [refer to D7 (DIO7 Configuration) parameter]. CTS is re-asserted after the DI Buffer has 34 bytes of memory available. **How to eliminate the need for flow control:**

1. Send messages that are smaller than the DI buffer size. 2. Interface at a lower baud rate [BD (Interface Data Rate) parameter] than the throughput data rate.

**Case in which the DI Buffer may become full and possibly overflow:**

If the module is receiving a continuous stream of RF data, any serial data that arrives on the DI pin is placed in the DI Buffer. The data in the DI buffer will be transmitted over-the-air when the module is no longer receiving RF data in the network.

Refer to the RO (Packetization Timeout), BD (Interface Data Rate) and D7 (DIO7 Configuration) command descriptions for more information.

**DO (Data Out) Buffer**

When RF data is received, the data enters the DO buffer and is sent out the serial port to a host device. Once the DO Buffer reaches capacity, any additional incoming RF data is lost. **Hardware Flow Control (RTS).** If RTS is enabled for flow control (D6 (DIO6 Configuration) Parameter = 1), data will not be sent out the DO Buffer as long as RTS (pin 16) is de-asserted. **Two cases in which the DO Buffer may become full and possibly overflow:**

1. If the RF data rate is set higher than the interface data rate of the module, the module will receive data from the transmitting module faster than it can send the data to the host.

]2. If the host does not allow the module to transmit data out from the DO buffer because of being held off by hardware or software flow control.

**4.2. ADC and Digital I/O Line Support**

The XBee/XBee-PRO RF Modules support ADC (Analog-to-digital conversion) and digital I/O line passing. The following pins support multiple functions: To enable ADC and DIO pin functions:

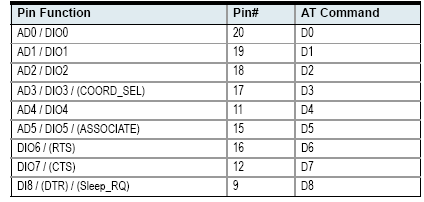


Table  4.2: Pin  functions  and  their  associated  pin  numbers  and  commands

AD  =  Analog‐to‐Digital  Converter,

DIO = Digital Input/Output Pin  functions  not  applicable  to  this  section  are  denoted  within  (parenthesis).

To enable ADC and DIO pin functions:]

For ADC Support: Set ATDn = 2

For Digital Input support: Set ATDn = 3

For Digital Output Low support: Set ATDn = 4

For Digital Output High support: Set ATDn = 5

**4.2.1. I/O Data Format**

I/O data begins with a header. The first byte of the header defines the number of samples forth-coming. A sample is comprised of input data and the inputs can contain either DIO or ADC. The last 2 bytes of the header (Channel Indicator) define which inputs are active. Each bit represents either a DIO line or ADC channel.

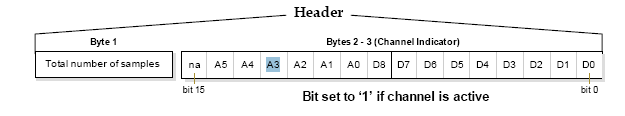


Figure 4.2.1(a): Header

Sample data follows the header and the channel indicator frame is used to determine how to read the sample data. If any of the DIO lines are enabled, the first 2 bytes are the DIO data and the ADC data follows. ADC channel data is stored as an unsigned 10-bit value right-justified on a 16- bit boundary.

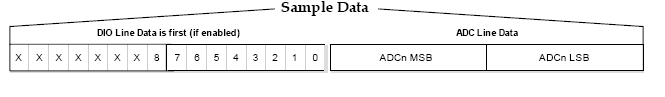


Figure  4.2.1(b):Sample  Data

**4.2.2. API Support**

I/O data is sent out the UART using an API frame. All other data can be sent and received using Transparent Operation [refer to p10] or API framing if API mode is enabled (AP > 0). API Operations support two RX (Receive) frame identifiers for I/O data: • 0x82 for RX (Receive) Packet: 64-bit address I/O • 0x83 for RX (Receive) Packet: 16-bit address I/O The API command header is the same as shown in the “RX (Receive) Packet: 64-bit Address” and “RX (Receive) Packet: 64-bit Address” API types [refer to p56]. RX data follows the format described in the I/O Data Format section [p12]. **Applicable Commands:** AP (API Enable)

**4.2.3. Sleep Support**

When an RF module wakes, it will always do a sample based on any active ADC or DIO lines. This allows sampling based on the sleep cycle whether it be Cyclic Sleep (SM parameter = 4 or 5) or Pin Sleep (SM = 1 or 2). To gather more samples when awake, set the IR (Sample Rate) parameter. For Cyclic Sleep modes: If the IR parameter is set, the module will stay awake until the IT (Samples before TX) parameter is met. The module will stay awake for ST (Time before Sleep) time. **Applicable Commands:** IR (Sample Rate), IT (Samples before TX), SM (Sleep Mode), IC (DIO Change Detect)

**4.2.4. DIO Pin Change Detect**

When “DIO Change Detect” is enabled (using the IC command), DIO lines 0-7 are monitored. When a change is detected on a DIO line, the following will occur:

1. An RF packet is sent with the updated DIO pin levels. This packet will not contain any ADC samples. 2. Any queued samples are transmitted before the change detect data. This may result in receiving a packet with less than IT (Samples before TX) samples.

Note: Change detect will not affect Pin Sleep wake-up. The D8 pin (DTR/Sleep RQ/DI8) is the only line that will wake a module from Pin Sleep. If not all samples are collected, the module will still enter Sleep Mode after a change detect packet is sent. **Applicable Commands**: IC (DIO Change Detect), IT (Samples before TX)

**4.2.5. Sample Rate (Interval)**

The Sample Rate (Interval) feature allows enabled ADC and DIO pins to be read periodically on modules that are not configured to operate in Sleep Mode. When one of the Sleep Modes is enabled and the IR (Sample Rate) parameter set, the module will stay awake until IT (Samples before TX) samples have been collected. Once a particular pin is enabled, the appropriate sample rate must be chosen. The maximum sample rate that can be achieved while using one A/D line is 1 sample/ms or 1 KHz (Note that the modem will not be able to keep up with transmission when IR & IT are equal to “1”). **Applicable Commands**: IR (Sample Rate), IT (Samples before TX), SM (Sleep Mode)

**4.2.6. I/O Line Passing**

Virtual wires can be set up between XBee/XBee-PRO Modules. When an RF data packet is received that contains I/O data, the receiving module can be setup to update any enabled outputs (PWM and DIO) based on the data it receives. Note that I/O lines are mapped in pairs. For example: AD0 can only update PWM0 and DI5 can only update DO5). The default setup is for outputs not to be updated, which results in the I/O data being sent out the UART (refer to the IU (Enable I/O Output) command). To enable the outputs to be updated, the IA (I/O Input Address) parameter must be setup with the address of the module that has the appropriate inputs enabled. This effectively binds the outputs to a particular module’s input. This does not affect the ability of the module to receive I/O line data from other modules - only its ability to update enabled outputs. The IA parameter can also be setup to accept I/O data for output changes from any module by setting the IA parameter to 0xFFFF. When outputs are changed from their non-active state, the module can be setup to return the out-put level to it non-active state. The timers are set using the Tn (Dn Output Timer) and PT (PWM Output Timeout) commands. The timers are reset every time a valid I/O packet (passed IA check) is received. The IC (Change Detect) and IR (Sample Rate) parameters can be setup to keep the output set to their active output if the system needs more time than the timers can handle.

Note: DI8 can not be used for I/O line passing.

**Applicable Commands:** IA (I/O Input Address), Tn (Dn Output Timeout), P0 (PWM0 Configuration), P1 (PWM1 Configuration), M0 (PWM0 Output Level), M1 (PWM1 Output Level), PT (PWM Output Timeout), RP (RSSSI PWM Timer)

4.2.7. Configuration Example

As an example for a simple A/D link, a pair of RF modules could be set as follows:

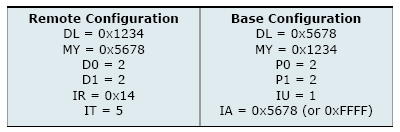


Table 4.2.7: Configuration Example

These settings configure the remote module to sample AD0 and AD1 once each every 20 ms.It then buffers 5 samples each before sending them back to the base module. The base should then receive a 32-Byte transmission (20 Bytes data and 12 Bytes framing) every 100 ms.

**4.3. XBee/XBee-PRO Addressing**

Every RF data packet sent over-the-air contains a Source Address and Destination Address field in its header. The RF module conforms to the 802.15.4 specification and supports both short 16-bit addresses and long 64-bit addresses. A unique 64-bit IEEE source address is assigned at the fac-tory and can be read with the SL (Serial Number Low) and SH (Serial Number High) commands. Short addressing must be configured manually. A module will use its unique 64-bit address as its Source Address if its MY (16-bit Source Address) value is “0xFFFF” or “0xFFFE”. To send a packet to a specific module using 64-bit addressing: Set Destination Address (DL + DH) to match the Source Address (SL + SH) of the intended destination module. To send a packet to a specific module using 16-bit addressing: Set DL (Destination Address Low) parameter to equal the MY parameter and set the DH (Destination Address High) parameter to ‘0’.

**4.3.1. Unicast Mode**

By default, the RF module operates in Unicast Mode. Unicast Mode is the only mode that supports retries. While in this mode, receiving modules send an ACK (acknowledgement) of RF packet reception to the transmitter. If the transmitting module does not receive the ACK, it will re-send the packet up to three times or until the ACK is received. **Short 16-bit addresses**. The module can be configured to use short 16-bit addresses as the Source Address by setting (MY < 0xFFFE). Setting the DH parameter (DH = 0) will configure the Destination Address to be a short 16-bit address (if DL < 0xFFFE). For two modules to communicate using short addressing, the Destination Address of the transmitter module must match the MY parameter of the receiver. The following table shows a sample network configuration that would enable Unicast Mode communications using short 16-bit addresses.

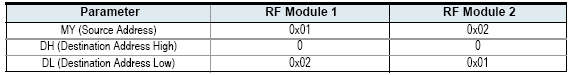


Table  4.3.1: Sample  Unicast  Network  Configuration  (using  16‐bit  addressing)

**Long 64-bit addresses**. The RF module’s serial number (SL parameter concatenated to the SH parameter) can be used as a 64-bit source address when the MY (16-bit Source Address) parameter is disabled. When the MY parameter is disabled (set MY = 0xFFFF or 0xFFFE), the module’s source address is set to the 64-bit IEEE address stored in the SH and SL parameters. When an End Device associates to a Coordinator, its MY parameter is set to 0xFFFE to enable 64- bit addressing. The 64-bit address of the module is stored as SH and SL parameters. To send a packet to a specific module, the Destination Address (DL + DH) on one module must match the Source Address (SL + SH) of the other.

**4.3.2. Broadcast Mode**

Any RF module within range will accept a packet that contains a broadcast address. When configured to operate in Broadcast Mode, receiving modules do not send ACKs (Acknowledgements) and transmitting modules do not automatically re-send packets as is the case in Unicast Mode. To send a broadcast packet to all modules regardless of 16-bit or 64-bit addressing, set the destination addresses of all the modules as shown below.

Sample Network Configuration (All modules in the network):

• DL (Destination Low Address) = 0x0000FFFF

• DH (Destination High Address) = 0x00000000 (default value)

NOTE: When programming the module, parameters are entered in hexadecimal notation (without the “0x” prefix). Leading zeros may be omitted.

**4.4 Modes of Operation**

XBee/XBee-PRO RF Modules operate in five modes.

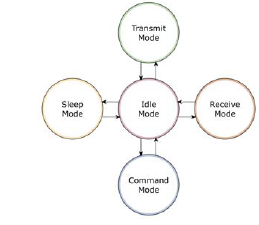
****

Figure  4.4.1. Modes  of  Operation

**4.4.1. Idle Mode**

When not receiving or transmitting data, the RF module is in Idle Mode. The module shifts into the other modes of operation under the following conditions:

• Transmit Mode (Serial data is received in the DI Buffer)

• Receive Mode (Valid RF data is received through the antenna)

• Sleep Mode (Sleep Mode condition is met)

• Command Mode (Command Mode Sequence is issued)

**4.4.2. Transmit/Receive Modes**

**RF Data Packets**

Each transmitted data packet contains a Source Address and Destination Address field. The Source Address matches the address of the transmitting module as specified by the MY (Source Address) parameter (if MY >= 0xFFFE), the SH (Serial Number High) parameter or the SL (Serial Number Low) parameter. The <Destination Address> field is created from the DH (Destination Address High) and DL (Destination Address Low) parameter values. The Source Address and/or Destination Address fields will either contain a 16-bit short or long 64-bit long address. The RF data packet structure follows the 802.15.4 specification. [Refer to the XBee/XBee-PRO Addressing section for more information]

**Direct and Indirect Transmission**

There are two methods to transmit data: • Direct Transmission - data is transmitted immediately to the Destination Address • Indirect Transmission - A packet is retained for a period of time and is only transmitted after the destination module (Source Address = Destination Address) requests the data. Indirect Transmissions can only occur on a Coordinator. Thus, if all nodes in a network are End Devices, only Direct Transmissions will occur. Indirect Transmissions are useful to ensure packet delivery to a sleeping node. The Coordinator currently is able to retain up to 2 indirect messages.

**Direct Transmission** ANonBeaconing Coordinator can be configured to use only Direct Transmission by setting the SP (Cyclic Sleep Period) parameter to “0”. Also, a NonBeaconing Coordinator using indirect transmissions will revert to direct transmission if it knows the destination module is awake. To enable this behavior, the ST (Time before Sleep) value of the Coordinator must be set to match the ST value of the End Device. Once the End Device either transmits data to the Coordinator or polls the Coordinator for data, the Coordinator will use direct transmission for all subsequent data transmissions to that module address until ST time (or number of beacons) occurs with no activity (at which point it will revert to using indirect transmissions for that module address). “No activity” means no transmission or reception of messages with a specific address. Global messages will not reset the ST timer.

**Indirect Transmission** To configure Indirect Transmissions in a PAN (Personal Area Network), the SP (Cyclic Sleep Period) parameter value on the Coordinator must be set to match the longest sleep value of any End Device. The SP parameter represents time in NonBeacon systems and beacons in Beacon-enabled systems. The sleep period value on the Coordinator determines how long (time or number of beacons) the Coordinator will retain an indirect message before discarding it. In NonBeacon networks, an End Device must poll the Coordinator once it wakes from Sleep to determine if the Coordinator has an indirect message for it. For Cyclic Sleep Modes, this is done automatically every time the module wakes (after SP time). For Pin Sleep Modes, the A1 (End Device Association) parameter value must be set to enable Coordinator polling on pin wake-up. Alternatively, an End Device can use the FP (Force Poll) command to poll the Coordinator as needed.

**CCA (Clear Channel Assessment)**

**Acknowledgement** Prior to transmitting a packet, a CCA (Clear Channel Assessment) is performed on the channel to determine if the channel is available for transmission. The detected energy on the channel is com-pared with the CA (Clear Channel Assessment) parameter value. If the detected energy exceeds the CA parameter value, the packet is not transmitted. Also, a delay is inserted before a transmission takes place. This delay is settable using the RN (Backoff Exponent) parameter. If RN is set to “0”, then there is no delay before the first CCA is per-formed. The RN parameter value is the equivalent of the “minBE” parameter in the 802.15.4 specification. The transmit sequence follows the 802.15.4 specification. By default, the MM (MAC Mode) parameter = 0. On a CCA failure, the module will attempt to re-send the packet up to two additional times. When in Unicast packets with RR (Retries) = 0, the module will execute two CCA retries. Broadcast packets always get two CCA retries.

If the transmission is not a broadcast message, the module will expect to receive an acknowledgement from the destination node. If an acknowledgement is not received, the packet will be resent up to 3 more times.

If the acknowledgement is not received after all transmissions, an ACK failure is recorded.

**4.4.3. Sleep Mode**

Sleep Modes enable the RF module to enter states of low-power consumption when not in use. In order to enter Sleep Mode, one of the following conditions must be met (in addition to the module having a non-zero SM parameter value): • Sleep\_RQ (pin 9) is asserted. • The module is idle (no data transmission or reception) for the amount of time defined by the ST (Time before Sleep) parameter. [NOTE: ST is only active when SM = 4-5.]

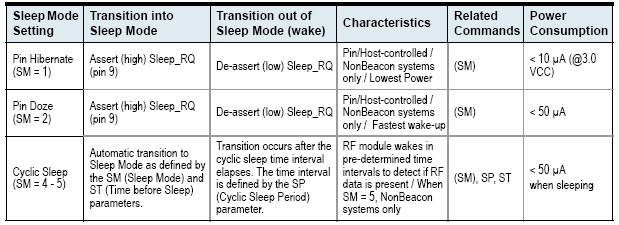
****

Table  4.4.3(a): Sleep  Mode  Configurations

The SM command is central to setting Sleep Mode configurations. By default, Sleep Modes are dis-abled (SM = 0) and the module remains in Idle/Receive Mode. When in this state, the module is constantly ready to respond to serial or RF activity. **Higher Voltages.**Sleep Mode current consumption is highly sensitive to voltage. Voltages above 3.0V will cause much higher current consumption

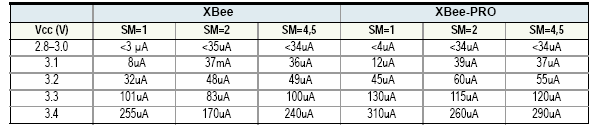
****

Table 4.4.3(b): Sample  Sleep  Mode  Currents

**Pin/Host-controlled Sleep Modes**

The transient current when waking from pin sleep (SM = 1 or 2) does not exceed the idle current of the module. The current ramps up exponentially to its idle current. **Pin Hibernate (SM = 1)** • Pin/Host-controlled • Typical power-down current: < 10 μA (@3.0 VCC) • Wake-up time: 13.2 msec Pin Hibernate Mode minimizes quiescent power (power consumed when in a state of rest or inac-tivity). This mode is voltage level-activated; when Sleep\_RQ is asserted, the module will finish any transmit, receive or association activities, enter Idle Mode and then enter a state of sleep. The module will not respond to either serial or RF activity while in pin sleep. To wake a sleeping module operating in Pin Hibernate Mode, de-assert Sleep\_RQ (pin 9). The module will wake when Sleep\_RQ is de-asserted and is ready to transmit or receive when the CTS line is low. When waking the module, the pin must be de-asserted at least two 'byte times” after CTS goes low. This assures that there is time for the data to enter the DI buffer.

**Pin Doze (SM = 2)** • Pin/Host-controlled • Typical power-down current: < 50 μA • Wake-up time: 2 msec Pin Doze Mode functions as does Pin Hibernate Mode; however, Pin Doze features faster wake-up time and higher power consumption. To wake a sleeping module operating in Pin Doze Mode, de-assert Sleep\_RQ (pin 9). The module will wake when Sleep\_RQ is de-asserted and is ready to transmit or receive when the CTS line is low. When waking the module, the pin must be de-asserted at least two 'byte times' after CTS goes low. This assures that there is time for the data to enter the DI buffer.

**Cyclic Sleep Modes**

**Cyclic Sleep Remote (SM = 4)** • Typical Power-down Current: < 50 μA (when asleep) • Wake-up time: 2 msec The Cyclic Sleep Modes allow modules to periodically check for RF data. When the SM parameter is set to ‘4’, the module is configured to sleep, then wakes once a cycle to check for data from a module configured as a Cyclic Sleep Coordinator (SM = 0, CE = 1). The Cyclic Sleep Remote sends a poll request to the coordinator at a specific interval set by the SP (Cyclic Sleep Period) parameter. The coordinator will transmit any queued data addressed to that specific remote upon receiving the poll request. If no data is queued for the remote, the coordinator will not transmit and the remote will return to sleep for another cycle. If queued data is transmitted back to the remote, it will stay awake to allow for back and forth communication until the ST (Time before Sleep) timer expires. Also note that CTS will go low each time the remote wakes, allowing for communication initiated by the remote host if desired.

**Cyclic Sleep Remote with Pin Wake-up (SM = 5)** Use this mode to wake a sleeping remote module through either the RF interface or by the de-assertion of Sleep\_RQ for event-driven communications. The cyclic sleep mode works as described above (Cyclic Sleep Remote) with the addition of a pin-controlled wake-up at the remote module. The Sleep\_RQ pin is edge-triggered, not level-triggered. The module will wake when a low is detected then set CTS low as soon as it is ready to transmit or receive. Any activity will reset the ST (Time before Sleep) timer so the module will go back to sleep only after there is no activity for the duration of the timer. Once the module wakes (pin-controlled), further pin activity is ignored. The module transitions back into sleep according to the ST time regardless of the state of the pin.

**[Cyclic Sleep Coordinator (SM = 6)]** • Typical current = Receive current • Always awake NOTE: The SM=6 parameter value exists solely for backwards compatibility with firmware version 1.x60. If backwards compatibility with the older firmware version is not required, always use the CE (Coordinator Enable) command to configure a module as a Coordinator. This mode configures a module to wake cyclic sleeping remotes through RF interfacing. The Coordinator will accept a message addressed to a specific remote 16 or 64-bit address and hold it in a buffer until the remote wakes and sends a poll request. Messages not sent directly (buffered and requested) are called "Indirect messages". The Coordinator only queues one indirect message at a time. The Coordinator will hold the indirect message for a period 2.5 times the sleeping period indicated by the SP (Cyclic Sleep Period) parameter. The Coordinator's SP parameter should be set to match the value used by the remotes.

**4.4.4. Command Mode**

**AT Command Mode** To modify or read RF Module parameters, the module must first enter into Command Mode - a state in which incoming characters are interpreted as commands. Two Command Mode options are supported: AT Command Mode [refer to section below] and API Command Mode [p52].

**To Enter AT Command Mode:**

Send the 3-character command sequence “+++” and observe guard times before and after the command characters. [Refer to the “Default AT Command Mode Sequence” below.]

Default AT Command Mode Sequence (for transition to Command Mode): • No characters sent for one second [GT (Guard Times) parameter = 0x3E8] • Input three plus characters (“+++”) within one second [CC (Command Sequence Character) parameter = 0x2B.] • No characters sent for one second [GT (Guard Times) parameter = 0x3E8] All of the parameter values in the sequence can be modified to reflect user preferences. NOTE: Failure to enter AT Command Mode is most commonly due to baud rate mismatch. Ensure the ‘Baud’ setting on the “PC Settings” tab matches the interface data rate of the RF module. By default, the BD parameter = 3 (9600 bps). **To Send AT Commands:**

Send AT commands and parameters using the syntax shown below.

****

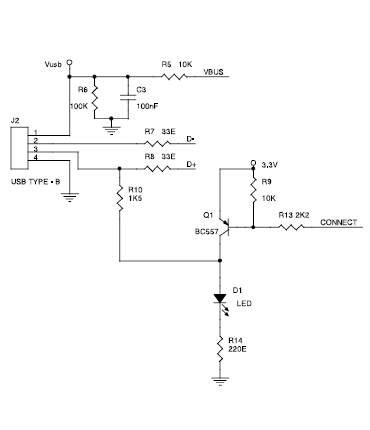
Figure 4.4.4.1:  Syntax  for  sending  AT  Commands

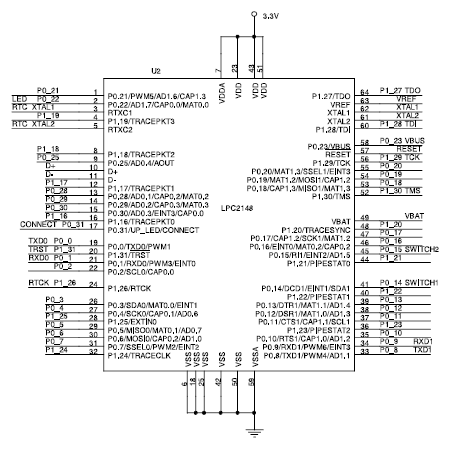
To read a parameter value stored in the RF module’s register, omit the parameter field. The preceding example would change the RF module Destination Address (Low) to “0x1F”. To store the new value to non-volatile (long term) memory, subsequently send the WR (Write) command. For modified parameter values to persist in the module’s registry after a reset, changes must be saved to non-volatile memory using the WR (Write) Command. Otherwise, parameters are restored to previously saved values after the module is reset. **System Response.**When a command is sent to the module, the module will parse and execute the command. Upon successful execution of a command, the module returns an “OK” message. If execution of a command results in an error, the module returns an “ERROR” message. **To Exit AT Command Mode:**

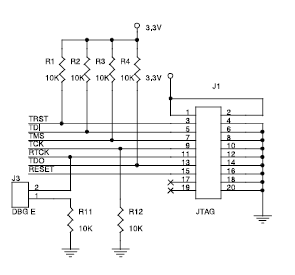
1. Send the ATCN (Exit Command Mode) command (followed by a carriage return). [OR] 2. If no valid AT Commands are received within the time specified by CT (Command Mode Timeout) Command, the RF module automatically returns to Idle Mode.

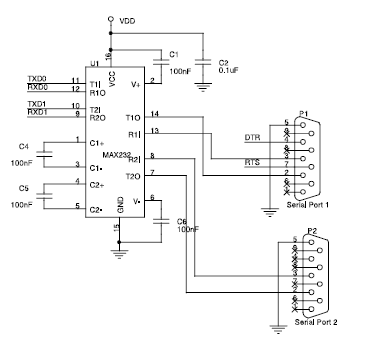
**Chapter 5. Schematic Representation**

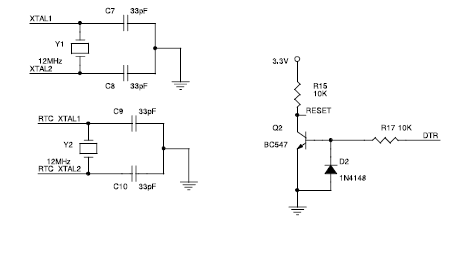
**5.1 Transmitter**

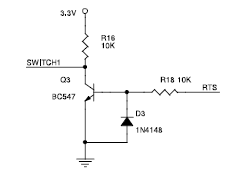
****

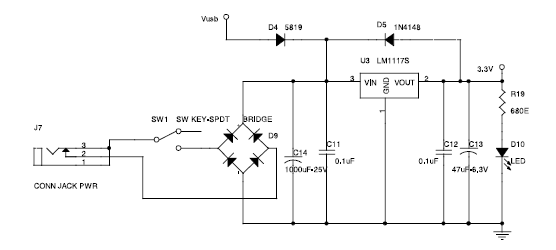
****

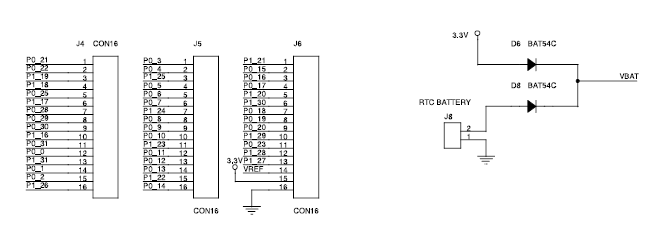
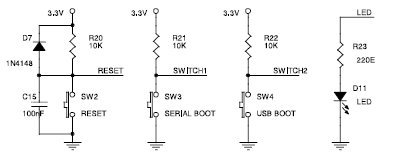
****

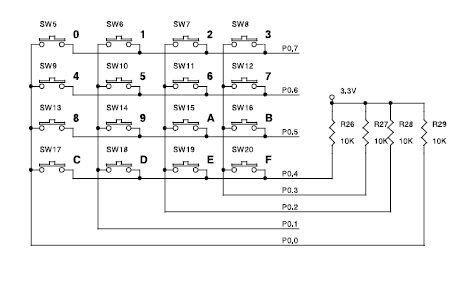
****

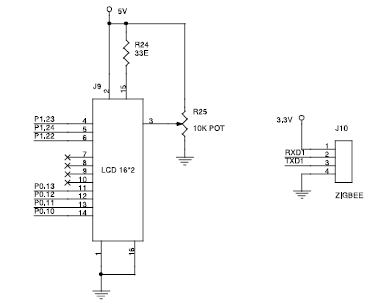
****

****

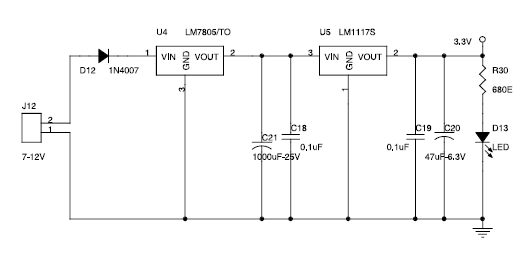
****

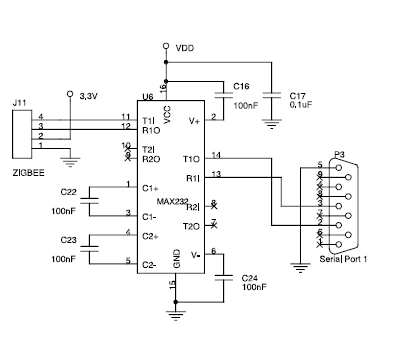
****

****

****

**5.2 Receiver**

****

****

**Chapter 6. Flow of the System**

6.1 Flow chart for LCD

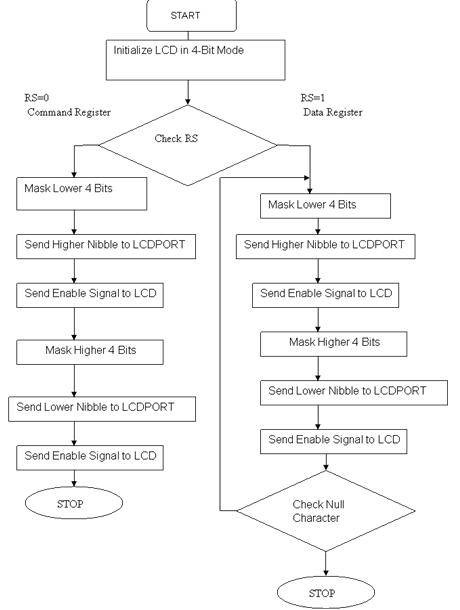
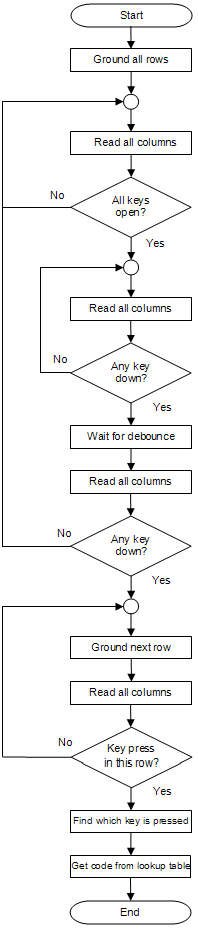


Figure 6.1.1: flow of the Lcd

6.2 Flow chart for Keypad



6.3 Flow chart of complete System

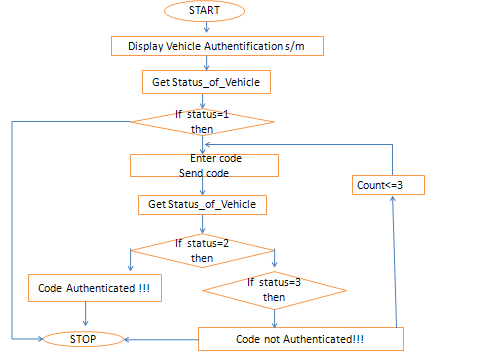


Figure 6.3.1: Flow of entire system

6.4 Using the Software

**Step-1.**Open the code in Microsoft visual basic.

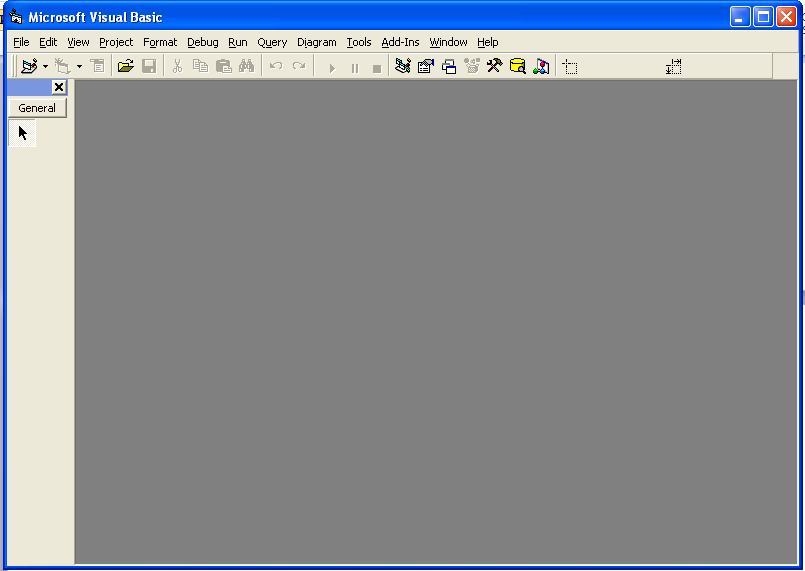
****

Figure 6.4.1: Microsoft visual basic window.

**Step-2.**Go to the run tab and select start with full compile.

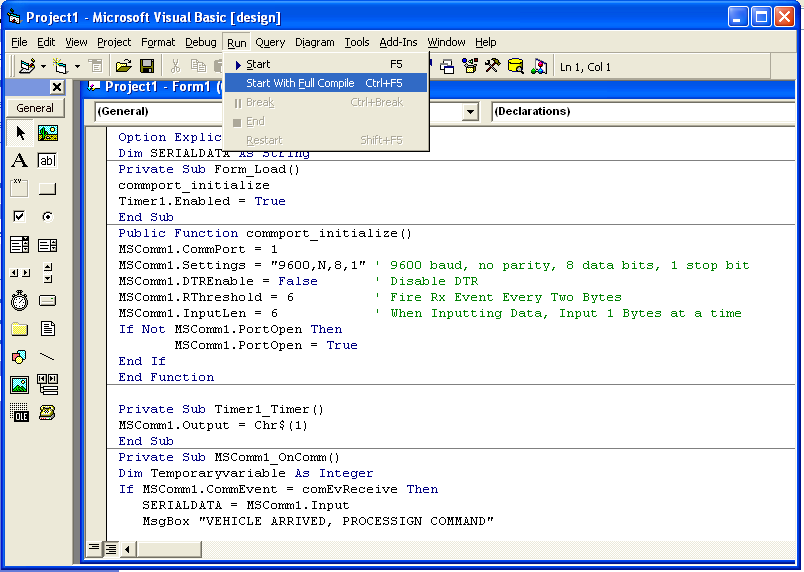


Figure 6.4.2: code compilation window.

**7. RESULT**

* ZIGBEE WIRLESS VEEHICULAR INDENTIFICATION AND AUTHENTICATION security based systems is a big break though in the security field.
* Minimize militant attacks like whatever had happened in the parliament, hotel Taj and hotel Oberio.

**8. CONCLUSION**

8.1 CONCLUSION

* The proposed ZigBee technology based Wireless Vehicle identification and driver authentication system satisfies the much needed requirement of vehicle security.
* Use of ZigBee module ensures the system to be low cost and low power consumable system.
* User-friendly interface and simple system setup.

8.2 Future Enhancement

* Of course after 5years our model(project idea)will become obsolete .
* After 5 years we can use our model with a small modification like at that time we will use GSM concept in place of ZIGBEE concept.
* After small changes what we will do. we will installed module1 in any government parking lot and module2 will installed inside the vehicle .
* Now when the vehicle enter at the entrance it will identified by the ZIGBEE transceiver installed at the gate.
* After that the GSM based system at the gate asks the driver to enter the password and this password is also verified by the system at the entrance gate.
* Now once both the vehicle identification is and the driver authentication has been done then the information is being sent by the system to the security guard at the gate.
* According to this information the guard will allows or denies access to the vehicle inside the premises.