Laboratory Journal
of
Embedded System Design(ESD)

For completion of term work of 8^{th} semester
curriculum program

Bachelor of Technology

in

ELECTRONICS AND TELECOMMUNICATION ENGINEERING

DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING
Dr. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY
Lonere-402 103, Tal. Mangaon, Dist. Raigad (MS)
INDIA
2011-2012

Contents

List of Figures

1 Experiment 1

1.1 Aim

To write a program in Embedded C for LED display Counter for LPC 2148 ARM processor.

1.2 Equipments And Software

Keil μ Vision for ARM, LPC 2148 development board,LEDs.

1.3 Theory

LPC2141/2/4/6/8 has two 32-bit General Purpose I/O ports(GPIO). Total of 30 input/output and a single output only pin out of 32 pins are available on PORT0. PORT1 has up to 16 pins available for GPIO functions. PORT0 and PORT1 are controlled via two groups of 4 registers. Legacy registers allow backward compatibility with earlier family devices, using existing code. The functions and relative timing of older GPIO implementations is preserved.

Features of GPIO

- Every physical GPIO port is accessible via either the group of registers providing an enhanced features and accelerated port access or the legacy group of registers.
- Accelerated GPIO functions:
  - GPIO registers are relocated to the ARM local bus so that the fastest possible I/O timing can be achieved.
  - Mask registers allow treating sets of port bits as a group, leaving other bits unchanged.
  - All registers are byte and half-word addressable.
  - Entire port value can be written in one instruction.

- Direction control of individual bits.
- Bit-level set and clear registers allow a single instruction set or clear of any number of bits in one port
- Backward compatibility with other earlier devices is maintained with legacy registers appearing at the original addresses on the APB bus.

Applications

- General purpose I/O.
- Driving LEDs, or other indicators.
- Controlling off-chip devices.
- Sensing digital inputs.

Registers and Pins

Name	Description	Access	Reset value[1]	Address
PINSEL0	Pin function select register 0.	Read/Write	0x0000 0000	0xE002 C000
PINSEL1	Pin function select register 1.	Read/Write	0x0000 0000	0xE002 C004
PINSEL2	Pin function select register 2.	Read/Write		0xE002 C014

[1] Reset value reflects the data stored in used bits only. It does not include reserved bits content.

Figure 1: Pin Connect Block Register Map

1.4 Pin function Select register 0 (PINSEL0 - 0xE002 C000)

Bit	Symbol	Value	Function	Reset value
1:0	P0.0	00	GPIO Port 0.0	0
		01	TXD (UART0)	
		10	PWM1	
		11	Reserved	
3:2	P0.1	00	GPIO Port 0.1	0
		01	RxD (UART0)	
		10	PWM3	
		11	EINT0	
5:4	P0.2	00	GPIO Port 0.2	0
		01	SCL0 (I^2C0)	
		10	Capture 0.0 (Timer 0)	
		11	Reserved	
7:6	P0.3	00	GPIO Port 0.3	0
		01	SDA0 (I^2C0)	
		10	Match 0.0 (Timer 0)	
		11	EINT1	
9:8	P0.4	00	GPIO Port 0.4	0
		01	SCK0 (SPI0)	
		10	Capture 0.1 (Timer 0)	
		11	AD0.6	
11:10	P0.5	00	GPIO Port 0.5	0
		01	MISO0 (SPI0)	
		10	Match 0.1 (Timer 0)	
		11	AD0.7	
13:12	P0.6	00	GPIO Port 0.6	0
		01	MOSI0 (SPI0)	
		10	Capture 0.2 (Timer 0)	
		11	Reserved[1][2] or AD1.0[3]	
15:14	P0.7	00	GPIO Port 0.7	0
		01	SSEL0 (SPI0)	
		10	PWM2	
		11	EINT2	
17:16	P0.8	00	GPIO Port 0.8	0
		01	TXD UART1	
		10	PWM4	
		11	Reserved[1][2] or AD1.1[3]	

Bit	Symbol	Value	Function	Reset value
1:0	P0.0	00	GPIO Port 0.0	0
		01	TXD (UART0)	
		10	PWM1	
		11	Reserved	
3:2	P0.1	00	GPIO Port 0.1	0
		01	RxD (UART0)	
		10	PWM3	
		11	EINT0	
5:4	P0.2	00	GPIO Port 0.2	0
		01	SCL0 (I^2C0)	
		10	Capture 0.0 (Timer 0)	
		11	Reserved	
7:6	P0.3	00	GPIO Port 0.3	0
		01	SDA0 (I^2C0)	
		10	Match 0.0 (Timer 0)	
		11	EINT1	
9:8	P0.4	00	GPIO Port 0.4	0
		01	SCK0 (SPI0)	
		10	Capture 0.1 (Timer 0)	
		11	AD0.6	
11:10	P0.5	00	GPIO Port 0.5	0
		01	MISO0 (SPI0)	
		10	Match 0.1 (Timer 0)	
		11	AD0.7	
13:12	P0.6	00	GPIO Port 0.6	0
		01	MOSI0 (SPI0)	
		10	Capture 0.2 (Timer 0)	
		11	Reserved[1][2] or AD1.0[3]	
15:14	P0.7	00	GPIO Port 0.7	0
		01	SSEL0 (SPI0)	
		10	PWM2	
		11	EINT2	
17:16	P0.8	00	GPIO Port 0.8	0
		01	TXD UART1	
		10	PWM4	
		11	Reserved[1][2] or AD1.1[3]	

Figure 2: Pin Function Select Register 0

The PINSEL1 register controls the functions of the pins as per the settings listed in following tables. The direction control bit in the IO0DIR register is effective only when the GPIO function is selected for a pin. For other functions direction is controlled automatically.

Conclusion

Hence we have studied the program for LED timer in LPC 2148

2 Experiment 2

2.1 Aim

To write a program in Embedded C for LCD interfacing with LPC 2148 ARM processor.

2.2 Equipments and softwares

Keil μ Vision for ARM, LPC 2148 development board,LCD.

2.3 Theory

Liquified Crystal Displays(LCDs) are most commonly used in modern embedded systems as display devices. LCDs can be used to display numbers as well as characters.The LCD commands are given in different ways. The LCDs can be interfaced with all 8 data lines as well as 4 data lines. The most commonly used Character based LCDs are based on Hitachi's HD44780 controller or other which are compatible with HD44580. In this tutorial, we will discuss about character based LCDs, their interfacing with various microcontrollers, various interfaces (8-bit/4-bit), programming, special stuff and tricks you can do with these simple looking LCDs which can give a new look to your application. The most commonly used LCDs found in the market today are 1 Line, 2 Line or 4 Line LCDs which have only 1 controller and support at most of 80 charachers, whereas LCDs supporting more than 80 characters make use of 2 HD44780 controllers.

Most LCDs with 1 controller has 14 Pins and LCDs with 2 controller has 16 Pins (two pins are extra in both for back-light LED connections). For our board, the LCD is interfaced in 8 bit mode. The port pins of GPIO used are P0.16 to P0.24 are used.

Command	Code										Description	Execution Time
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears the display and returns the cursor to the home position (address 0).	82μs~1.64ms
Return Home	0	0	0	0	0	0	0	0	1	*	Returns the cursor to the home position (address 0). Also returns a shifted display to the home position. DD RAM contents remain unchanged.	40μs~1.64ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and enables/disables the display.	40μs
Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B	Turns the display ON/OFF (D), or the cursor ON/OFF (C), and blink of the character at the cursor position (B).	40μs
Cursor & Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing the DD RAM contents.	40μs
Function Set	0	0	0	0	1	DL	N$	F	*	#	Sets the data width (DL), the number of lines in the display (L), and the character font (F).	40μs
Set CG RAM Address	0	0	0	1	A_{CG}						Sets the CG RAM address. CG RAM data can be read or altered after making this setting.	40μs
Set DD RAM Address	0	0	1	A_{DD}							Sets the DD RAM address. Data may be written or read after making this setting.	40μs
Read Busy Flag & Address	0	1	BF	AC							Reads the BUSY flag (BF) indicating that an internal operation is being performed and reads the address counter contents.	1μs
Write Data to CG or DD RAM	1	0	Write Data								Writes data into DD RAM or CG RAM.	46μs
Read Data from CG or DD RAM	1	1	Read Data								Reads data from DD RAM or CG RAM.	46μs
	I/D = 1: Increment I/D = 0: Decrement S = 1: Accompanies display shift. S/C= 1: Display shift S/C = 0: cursor move R/L= 1: Shift to the right. R/L= 0: Shift to the left. DL = 1: 8 bits DL = 0: 4 bits N = 1: 2 lines N = 0: 1 line F = 1: 5x10 dots F = 0: 5 x 7 dots BF = 1: Busy BF = 0: Can accept data *# Set to 1 on 24x4 modules* *$ With KS0072 is Address Mode.*										DD RAM: Display data RAM CG RAM: Character generator RAM A_{CG}: CG RAM Address A_{DD}: DD RAM Address Corresponds to cursor address. AC: Address counter Used for both DD and CG RAM address.	Execution times are typical. If transfers are timed by software and the busy flag is not used, add 10% to the above times.

Figure 3: LCD functioning codes

LCD commands

No.	Instruction	Hex	Decimal
1	Function Set: 8-bit, 1 Line, 5x7 Dots	0x30	48
2	Function Set: 8-bit, 2 Line, 5x7 Dots	0x38	56
3	Function Set: 4-bit, 1 Line, 5x7 Dots	0x20	32
4	Function Set: 4-bit, 2 Line, 5x7 Dots	0x28	40
5	Entry Mode	0x06	6
6	Display off Cursor off (clearing display without clearing DDRAM content)	0x08	8
7	Display on Cursor on	0x0E	14
8	Display on Cursor off	0x0C	12
9	Display on Cursor blinking	0x0F	15
10	Shift entire display left	0x18	24
12	Shift entire display right	0x1C	30
13	Move cursor left by one character	0x10	16
14	Move cursor right by one character	0x14	20
15	Clear Display (also clear DDRAM content)	0x01	1
16	Set DDRAM address or coursor position on display	0x80+add*	128+add*
17	Set CGRAM address or set pointer to CGRAM location	0x40+add**	64+add**

Figure 4: LCD Commands

Conclusion

Thus we have studied interfacing LCD to LPC 2148.

3 Experiment 3

3.1 Aim

To write a program in Embedded C for Digital to Analog Conversion using on chip DAC of LPC 2148 ARM processor.

3.2 Equipments and softwares

Keil μ Vision for ARM, LPC 2148 development board,LCD.

3.3 Theory

In electronics, a digital-to-analog converter (DAC or D-to-A) is a device that converts a digital (usually binary) code to an analog signal (current, voltage, or electric charge). An analog-to-digital converter (ADC) performs the reverse operation. Signals are easily stored and transmitted in digital form, but a DAC is needed for the signal to be recognized by human senses or other non-digital systems. A common use of digital-to-analog converters is generation of audio signals from digital information in music players. Digital video signals are converted to analog in televisions and cell phones to display colors and shades. Digital-to-analog conversion can degrade a signal, so conversion details are normally chosen so that the errors are negligible. Due to cost and the need for matched components, DACs are almost exclusively manufactured on integrated circuits (ICs). There are many DAC architectures which have different advantages and disadvantages. The suitability of a particular DAC for an application is determined by a variety of measurements including speed and resolution.

A DAC converts an abstract finite-precision number (usually a fixed-point binary number) into a physical quantity (e.g., a voltage or a pressure). In particular, DACs are often used to convert finite-precision time series data to a continually varying physical signal. A typical DAC converts the abstract numbers into a concrete sequence of impulses that are then processed by a reconstruction filter using some form of interpolation to fill in data between the impulses. Other DAC methods (e.g., methods based on delta-sigma

modulation) produce a pulse-density modulated signal that can then be filtered in a similar way to produce a smoothly varying signal. As per the NyquistShannon sampling theorem, a DAC can reconstruct the original signal from the sampled data provided that its bandwidth meets certain requirements (e.g., a baseband signal with bandwidth less than the Nyquist frequency). Digital sampling introduces quantization error that manifests as low-level noise added to the reconstructed signal.

This peripheral is available on chip in LPC2142/4/6/8 devices.

- 10 bit digital to analog converter
- Resistor string architecture
- Buffered output
- Power-down mode
- Selectable speed vs. power

Pin Description

Pin	Type	Description
AOUT	Output	**Analog Output.** After the selected settling time after the DACR is written with a new value, the voltage on this pin (with respect to V_{SSA}) is VALUE/1024 * V_{REF}.
V_{REF}	Reference	**Voltage Reference.** This pin provides a voltage reference level for the D/A converter.
V_{DDA}, V_{SSA}	Power	**Analog Power and Ground.** These should be nominally the same voltages as V_3 and V_{SSD}, but should be isolated to minimize noise and error.

Figure 5: Pin Description

DAC Register (DACR - 0xE006 C000)

This read/write register includes the digital value to be converted to analog, and a bit that trades off performance vs. power. Bits 5:0 are reserved for future, higher-resolution D/A converters. Bits 19:18 of the PINSEL1 register control whether the DAC is enabled

Bit	Symbol	Value	Description	Reset value
5:0	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
15:6	VALUE		After the selected settling time after this field is written with a new VALUE, the voltage on the A_{OUT} pin (with respect to V_{SSA}) is VALUE/1024 × V_{REF}.	0
16	BIAS	0	The settling time of the DAC is 1 μs max, and the maximum current is 700 μA.	0
		1	The settling time of the DAC is 2.5 μs and the maximum current is 350 μA.	
31:17	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Figure 6: DAC Register

and controlling the state of pin P0.25/AD0.4/AOUT. When these bits are 10, the DAC is powered on and active. The settling times noted in the description of the BIAS bit are valid for a capacitance load on the AOUT pin not exceeding 100 pF. A load impedance value greather than that value will cause settling time longer than the specified time.

Conclusion

Hence we have studied the onchip DAC of LPC 2148 ARM.

4 Experiment 4

4.1 Aim

To write a program in Embedded C for Analog to Digital Conversion using on chip ADC of LPC 2148 ARM processor.

4.2 Equipments and softwares

Keil μ Vision for ARM, LPC 2148 development board.

4.3 Theory

An analog-to-digital converter (abbreviated ADC, A/D or A to D) is a device that converts a continuous quantity to a discrete time digital representation. An ADC may also provide an isolated measurement. The reverse operation is performed by a digital-to-analog converter (DAC). Typically, an ADC is an electronic device that converts an input analog voltage or current to a digital number proportional to the magnitude of the voltage or current. However, some non-electronic or only partially electronic devices, such as rotary encoders, can also be considered ADCs. The digital output may use different coding schemes. Typically the digital output will be a two's complement binary number that is proportional to the input, but there are other possibilities. An encoder, for example, might output a Gray code. Basic clocking for the A/D converters is provided by the APB clock. A programmable divider is included in each converter, to scale this clock to the 4.5 MHz (max) clock needed by the successive approximation process. A fully accurate conversion requires 11 of these clocks.

- 10 bit successive approximation analog to digital converter (one in LPC2141/2 and two in LPC2144/6/8).
- Input multiplexing among 6 or 8 pins (ADC0 and ADC1).
- Power-down mode.

- Measurement range 0 V to VREF (typically 3 V; not to exceed VDDA voltage level).
- 10 bit conversion time = 2.44 s.
- Burst conversion mode for single or multiple inputs.

ADC Pin description and Register

Pin	Type	Description
AD0.7:6, AD0.4:1 & AD1.7:0 (LPC2144/6/8)	Input	**Analog Inputs.** The ADC cell can measure the voltage on any of these input signals. Note that these analog inputs are always connected to their pins, even if the Pin function Select register assigns them to port pins. A simple self-test of the ADC can be done by driving these pins as port outputs. **Note:** if the ADC is used, signal levels on analog input pins must not be above the level of V_{3A} at any time. Otherwise, A/D converter readings will be invalid. If the A/D converter is not used in an application then the pins associated with A/D inputs can be used as 5 V tolerant digital IO pins. **Warning:** while the ADC pins are specified as 5 V tolerant (see Section 5.2 "Pin description for LPC2141/2/4/6/8" on page 52), the analog multiplexing in the ADC block is not. More than 3.3 V (V_{DDA}) should not be applied to any pin that is selected as an ADC input, or the ADC reading will be incorrect. If for example AD0.0 and AD0.1 are used as the ADC0 inputs and voltage on AD0.0 = 4.5 V while AD0.1 = 2.5 V, an excessive voltage on the AD0.0 can cause an incorrect reading of the AD0.1, although the AD0.1 input voltage is within the right range.
V_{REF}	Reference	**Voltage Reference.** This pin is provides a voltage reference level for the A/D converter(s).
V_{DDA}, V_{SSA}	Power	**Analog Power and Ground.** These should be nominally the same voltages as V_{DD} and V_{SS}, but should be isolated to minimize noise and error.

Figure 7: ADC Pin Description

Generic Name	Description	Access	Reset value[1]	AD0 Address & Name	AD1 Address & Name
ADCR	A/D Control Register. The ADCR register must be written to select the operating mode before A/D conversion can occur.	R/W	0x0000 0001	0xE003 4000 AD0CR	0xE006 0000 AD1CR
ADGDR	A/D Global Data Register. This register contains the ADC's DONE bit and the result of the most recent A/D conversion.	R/W	NA	0xE003 4004 AD0GDR	0xE006 0004 AD1GDR
ADSTAT	A/D Status Register. This register contains DONE and OVERRUN flags for all of the A/D channels, as well as the A/D interrupt flag.	RO	0x0000 0000	0xE003 4030 AD0STAT	0xE006 0030 AD1STAT
ADGSR	A/D Global Start Register. This address can be written (in the AD0 address range) to start conversions in both A/D converters simultaneously.	WO	0x00	0xE003 4008 ADGSR	
ADINTEN	A/D Interrupt Enable Register. This register contains enable bits that allow the DONE flag of each A/D channel to be included or excluded from contributing to the generation of an A/D interrupt.	R/W	0x0000 0100	0xE003 400C AD0INTEN	0xE006 000C AD1INTEN
ADDR0	A/D Channel 0 Data Register. This register contains the result of the most recent conversion completed on channel 0.	RO	NA	0xE003 4010 AD0DR0	0xE006 0010 AD1DR0
ADDR1	A/D Channel 1 Data Register. This register contains the result of the most recent conversion completed on channel 1.	RO	NA	0xE003 4014 AD0DR1	0xE006 0014 AD1DR1
ADDR2	A/D Channel 2 Data Register. This register contains the result of the most recent conversion completed on channel 2.	RO	NA	0xE003 4018 AD0DR2	0xE006 0018 AD1DR2
ADDR3	A/D Channel 3 Data Register. This register contains the result of the most recent conversion completed on channel 3.	RO	NA	0xE003 401C AD0DR3	0xE006 001C AD1DR3
ADDR4	A/D Channel 4 Data Register. This register contains the result of the most recent conversion completed on channel 4.	RO	NA	0xE003 4020 AD0DR4	0xE006 0020 AD1DR4
ADDR5	A/D Channel 5 Data Register. This register contains the result of the most recent conversion completed on channel 5.	RO	NA	0xE003 4024 AD0DR5	0xE006 0024 AD1DR5
ADDR6	A/D Channel 6 Data Register. This register contains the result of the most recent conversion completed on channel 6.	RO	NA	0xE003 4028 AD0DR6	0xE006 0028 AD1DR6
ADDR7	A/D Channel 7 Data Register. This register contains the result of the most recent conversion completed on channel 7.	RO	NA	0xE003 402C AD0DR7	0xE006 002C AD1DR7

[1] Reset value reflects the data stored in used bits only. It does not include reserved bits content.

Figure 8: ADC registers

Bit	Symbol	Value	Description	Reset value
7:0	SEL		Selects which of the AD0.7:0/AD1.7:0 pins is (are) to be sampled and converted. For AD0, bit 0 selects Pin AD0.0, and bit 7 selects pin AD0.7. In software-controlled mode, only one of these bits should be 1. In hardware scan mode, any value containing 1 to 8 ones. All zeroes is equivalent to 0x01.	0x01
15:8	CLKDIV		The APB clock (PCLK) is divided by (this value plus one) to produce the clock for the A/D converter, which should be less than or equal to 4.5 MHz. Typically, software should program the smallest value in this field that yields a clock of 4.5 MHz or slightly less, but in certain cases (such as a high-impedance analog source) a slower clock may be desirable.	0
16	BURST	1	The AD converter does repeated conversions at the rate selected by the CLKS field, scanning (if necessary) through the pins selected by 1s in the SEL field. The first conversion after the start corresponds to the least-significant 1 in the SEL field, then higher numbered 1-bits (pins) if applicable. Repeated conversions can be terminated by clearing this bit, but the conversion that's in progress when this bit is cleared will be completed. **Remark:** START bits must be 000 when BURST = 1 or conversions will not start.	0
		0	Conversions are software controlled and require 11 clocks.	
19:17	CLKS		This field selects the number of clocks used for each conversion in Burst mode, and the number of bits of accuracy of the result in the RESULT bits of ADDR, between 11 clocks (10 bits) and 4 clocks (3 bits).	000
		000	11 clocks / 10 bits	
		001	10 clocks / 9bits	
		010	9 clocks / 8 bits	
		011	8 clocks / 7 bits	
		100	7 clocks / 6 bits	
		101	6 clocks / 5 bits	
		110	5 clocks / 4 bits	
		111	4 clocks / 3 bits	
20	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
21	PDN	1	The A/D converter is operational.	0
		0	The A/D converter is in power-down mode.	
23:22	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Figure 9: ADC registers

4.4 Conclusion

Hence we have studied the onchip ADC of LPC 2148 ARM.

5 Experiment 5

5.1 Aim

To write a program in Embedded C for using on chip Universal Asynchronous Receiver Transmitter(UART) of LPC 2148 ARM processor.

5.2 Equipments and softwares

Keil μ Vision for ARM, LPC 2148 development board.

5.3 Theory

The Universal Asynchronous Receiver/Transmitter (UART) takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. Each UART contains a shift register, which is the fundamental method of conversion between serial and parallel forms. Serial transmission of digital information (bits) through a single wire or other medium is much more cost effective than parallel transmission through multiple wires. The UART usually does not directly generate or receive the external signals used between different items of equipment. Separate interface devices are used to convert the logic level signals of the UART to and from the external signaling levels. External signals may be of many different forms. Examples of standards for voltage signaling are RS-232, RS-422 and RS-485 from the EIA. Historically, current (in current loops) was used in telegraph circuits. Some signaling schemes do not use electrical wires. Examples of such are optical fiber, IrDA (infrared), and (wireless) Bluetooth in its Serial Port Profile (SPP). Some signaling schemes use modulation of a carrier signal (with or without wires). Examples are modulation of audio signals with phone line modems, RF modulation with data radios, and the DC-LIN for power line communication. Communication may be simplex (in one direction only, with no provision for the receiving device to send information back to the transmitting device), full duplex (both devices send and receive at the same time) or half duplex (devices take turns transmitting and receiving).

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 with autobauding capabilities.
- Mechanism that enables software and hardware flow control implementation.

Pin Description and Registers

Pin	Type	Description
RXD0	Input	**Serial Input.** Serial receive data.
TXD0	Output	**Serial Output.** Serial transmit data.

Figure 10: UART pins

UART0 contains registers organized as shown below. The Divisor Latch Access Bit (DLAB) is contained in U0LCR[7] and enables access to the Divisor Latches.

Name	Description	Bit functions and addresses								Access	Reset value[1]	Address
		MSB							LSB			
		BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0			
U0RBR	Receiver Buffer Register	8-bit Read Data								RO	NA	0xE000 C000 (DLAB=0)
U0THR	Transmit Holding Register	8-bit Write Data								WO	NA	0xE000 C000 (DLAB=0)
U0DLL	Divisor Latch LSB	8-bit Data								R/W	0x01	0xE000 C000 (DLAB=1)
U0DLM	Divisor Latch MSB	8-bit Data								R/W	0x00	0xE000 C004 (DLAB=1)
U0IER	Interrupt Enable Register	-	-	-	-	-	-	En.ABTO	En.ABEO	R/W	0x00	0xE000 C004 (DLAB=0)
		-	-	-	-	-	En.RX Lin.St.Int	Enable THRE Int	En. RX Dat.Av.Int			
U0IIR	Interrupt ID Reg.	-	-	-	-	-	-	ABTO Int	ABEO Int	RO	0x01	0xE000 C008
		FIFOs Enabled		-	-	IIR3	IIR2	IIR1	IIR0			
U0FCR	FIFO Control Register	RX Trigger		-	-	-	TX FIFO Reset	RX FIFO Reset	FIFO Enable	WO	0x00	0xE000 C008
U0LCR	Line Control Register	DLAB	Set Break	Stick Parity	Even Par.Selct.	Parity Enable	No. of Stop Bits	Word Length Select		R/W	0x00	0xE000 C00C
U0LSR	Line Status Register	RX FIFO Error	TEMT	THRE	BI	FE	PE	OE	DR	RO	0x60	0xE000 C014
U0SCR	Scratch Pad Reg.	8-bit Data								R/W	0x00	0xE000 C01C
U0ACR	Auto-baud Control Register	-	-	-	-	-	-	ABTO Int.Clr	ABEO Int.Clr	R/W	0x00	0xE000 C020
		-	-	-	-	-	Aut.Rstrt.	Mode	Start			
U0FDR	Fractional Divider Register	Reserved[31:8]									0x10	0xE000 C028
		MulVal					DivAddVal					
U0TER	TX. Enable Reg.	TXEN	-	-	-	-	-	-	-	R/W	0x80	0xE000 C030

[1] Reset value reflects the data stored in used bits only. It does not include reserved bits content.

Figure 11: UART registers

Conclusion

Hence we studied the serial communication using UART0 on LPC 2148.

6 Experiment 6

6.1 Aim

To write a program in Embedded C for Keyboard interfacing with LPC 2148 ARM processor.

6.2 Equipments and softwares

Keil μ Vision for ARM, LPC 2148 development board,LCD.

6.3 Theory

Keypads are a part of HMI or Human Machine Interface and play really important role in a small embedded system where human interaction or human input is needed. Martix keypads are well known for their simple architecture and ease of interfacing with any microcontroller.

Matrix Keypad

Constuction of a keypad is really simple. As per the outline shown in the figure below we have four rows and four columns. In between each overlapping row and column line there is a key.

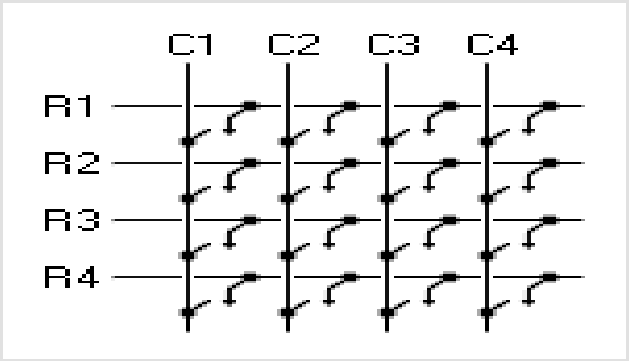

Figure 12: Keyboard schematics

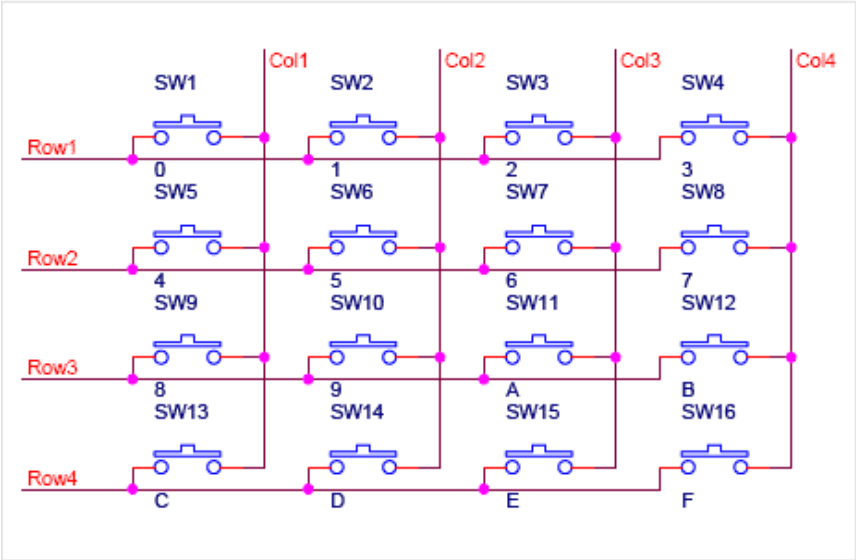

Figure 13: Keyboard Connections

There are many methods depending on how you connect your keypad with your controller, but the basic logic is same. We make the coloums as i/p and we drive the rows making them o/p, this whole procedure of reading the keyboard is called scanning.

In order to detect which key is pressed from the matrix, we make row lines low one by one and read the coloums. Lets say we first make Row1 low, then read the columns. If any of the key in row1 is pressed will make the corrosponding column as low i.e if second key is pressed in Row1, then column2 will give low. So we come to know that key 2 of Row1 is pressed. This is how scanning is done.

So to scan the keypad completely, we need to make rows low one by one and read the columns. If any of the button is pressed in a row, it will take the corresponding column to a low state which tells us that a key is pressed in that row. If button 1 of a row is pressed then Column 1 will become low, if button 2 then column2 and so on...

Conclusion

Hence we have studied keyboard interfacing with LPC 2148.

7 Experiment 7

7.1 Aim

Case study of an Embedded System: Digital Camera

7.2 Theory

In the past twenty years, most of the major technological breakthroughs in consumer electronics have really been part of one larger breakthrough. When you get down to it, CDs, DVDs, HDTV, MP3s and DVRs are all built around the same basic process: converting conventional analog information (represented by a fluctuating wave) into digital information (represented by ones and zeros, or bits). This fundamental shift in technology totally changed how we handle visual and audio information,it completely redefined what is possible. Conventional cameras depend entirely on chemical and mechanical processes ,you don't even need electricity to operate them. On the other hand, all digital cameras have a built-in computer, and all of them record images electronically.The new approach has been enormously successful. Since film still provides better picture quality, digital cameras have not completely replaced conventional cameras. But, as digital imaging technology has improved, digital cameras have rapidly become more popular.

Digital Camera Basics

A digital camera (or digicam) is a camera that takes video or still photographs, or both, digitally by recording images via an electronic image sensor. It is the main device used in the field of digital photography. Most 21st century cameras are digital. Digital cameras can do things film cameras cannot: displaying images on a screen immediately after they are recorded, storing thousands of images on a single small memory device, and deleting images to free storage space. The majority, including most compact cameras, can record moving video with sound as well as still photographs. Some can crop and stitch pictures and perform other elementary image editing. Some have a GPS receiver built in, and can produce geotagged photographs. The optical system works the same

as in film cameras, typically using a lens with a variable diaphragm to focus light onto an image pickup device. The diaphragm and shutter admit the correct amount of light to the imager, just as with film but the image pickup device is electronic rather than chemical. Most digicams, apart from camera phones and a few specialized types, have a standard tripod screw. Digital cameras are incorporated into many devices ranging from PDAs and mobile phones (called camera phones) to vehicles. The Hubble Space Telescope and other astronomical devices are essentially specialized digital cameras.

Image Resolution

The resolution of a digital camera is often limited by the image sensor (typically a CCD or CMOS sensor chip) that turns light into discrete signals, replacing the job of film in traditional photography. The sensor is made up of millions of "buckets" that essentially count the number of photons that strike the sensor. This means that the brighter the image at a given point on the sensor, the larger the value that is read for that pixel. Depending on the physical structure of the sensor, a color filter array may be used which requires a demosaicing/interpolation algorithm. The number of resulting pixels in the image determines its "pixel count". For example, a 640x480 image would have 307,200 pixels, or approximately 307 kilopixels; a 3872x2592 image would have 10,036,224 pixels, or approximately 10 megapixels.

The pixel count alone is commonly presumed to indicate the resolution of a camera, but this simple figure of merit is a misconception. Other factors impact a sensor's resolution, including sensor size, lens quality, and the organization of the pixels (for example, a monochrome camera without a Bayer filter mosaic has a higher resolution than a typical color camera). Where such other factors limit the resolution, a greater pixel count does not improve it, but may rather make the digital images inconveniently large and/or exacerbate image noise. Many digital compact cameras are criticized for having excessive pixels. Sensors can be so small that their 'buckets' can easily overfill; again, resolution of a sensor can become greater than the camera lens could possibly deliver.

As the technology has improved, costs have decreased dramatically. Counting the "pixels per dollar" as a basic measure of value for a digital camera, there has been a

continuous and steady increase in the number of pixels each dollar buys in a new camera, in accord with the principles of Moore's Law. This predictability of camera prices was first presented in 1998 at the Australian PMA DIMA conference by Barry Hendy and since referred to as "Hendy's Law". Since only a few aspect ratios are commonly used (mainly 4:3 and 3:2), the number of sensor sizes that are useful is limited. Furthermore, sensor manufacturers do not produce every possible sensor size, but take incremental steps in sizes. For example, in 2007 the three largest sensors (in terms of pixel count) used by Canon were the 21.1, 17.9, and 16.6 megapixel CMOS sensors. Demanding high quality and resolution (e.g. for use in professional photography), this count is an object of manufacturer competition. The highest resolution available on the market for consumer digital cameras is 80.1 MP.

Method of Image Capture

Since the first digital backs were introduced, there have been three main methods of capturing the image, each based on the hardware configuration of the sensor and color filters. The first method is often called single-shot, in reference to the number of times the camera's sensor is exposed to the light passing through the camera lens. Single-shot capture systems use either one CCD with a Bayer filter mosaic, or three separate image sensors (one each for the primary additive colors red, green, and blue) which are exposed to the same image via a beam splitter. The second method is referred to as multi-shot because the sensor is exposed to the image in a sequence of three or more openings of the lens aperture. There are several methods of application of the multi-shot technique. The most common originally was to use a single image sensor with three filters (once again red, green and blue) passed in front of the sensor in sequence to obtain the additive color information. Another multiple shot method is called Microscanning. This technique utilizes a single CCD with a Bayer filter but actually moved the physical location of the sensor chip on the focus plane of the lens to "stitch" together a higher resolution image than the CCD would allow otherwise. A third version combined the two methods without a Bayer filter on the chip. The third method is called scanning because the sensor moves across the focal plane much like the sensor of a desktop scanner. Their linear or tri-linear

sensors utilize only a single line of photosensors, or three lines for the three colors. In some cases, scanning is accomplished by moving the sensor e.g. when using Color co-site sampling or rotate the whole camera; a digital rotating line camera offers images of very high total resolution. The choice of method for a given capture is determined largely by the subject matter. It is usually inappropriate to attempt to capture a subject that moves with anything but a single-shot system. However, the higher color fidelity and larger file sizes and resolutions available with multi-shot and scanning backs make them attractive for commercial photographers working with stationary subjects and large-format photographs. Dramatic improvements in single-shot cameras and raw image file processing at the beginning of the 21st century made single shot, CCD-based cameras almost completely dominant, even in high-end commercial photography. CMOS-based single shot cameras remained somewhat common.

Filter mosaics, interpolation, and aliasing

Most current consumer digital cameras use a Bayer filter mosaic in combination with an optical anti-aliasing filter to reduce the aliasing due to the reduced sampling of the different primary-color images. A demosaicing algorithm is used to interpolate color information to create a full array of RGB image data. Cameras that use a beam-splitter single-shot 3CCD approach, three-filter multi-shot approach, Color co-site sampling or Foveon X3 sensor do not use anti-aliasing filters, nor demosaicing. Firmware in the camera, or a software in a raw converter program such as Adobe Camera Raw, interprets the raw data from the sensor to obtain a full color image, because the RGB color model requires three intensity values for each pixel: one each for the red, green, and blue (other color models, when used, also require three or more values per pixel). A single sensor element cannot simultaneously record these three intensities, and so a color filter array (CFA) must be used to selectively filter a particular color for each pixel. The Bayer filter pattern is a repeating 22 mosaic pattern of light filters, with green ones at opposite corners and red and blue in the other two positions. The high proportion of green takes advantage of properties of the human visual system, which determines brightness mostly from green and is far more sensitive to brightness than to hue or saturation. Sometimes

a 4-color filter pattern is used, often involving two different hues of green. This provides potentially more accurate color, but requires a slightly more complicated interpolation process. The color intensity values not captured for each pixel can be interpolated (or guessed) from the values of adjacent pixels which represent the color being calculated

7.3 Conclusion

Hence we studied the working of Digital Camera.