

This page is intentionally kept blank to preserve page format.

Index

Sl	Name of Experiment	Date	Remarks
	Vision and Mission statement		
1	Safety Procedures		
2	Handling electrostatic discharge (ESD)		
3	Familiarization of AVR studio 6		
4	Study of ports		
5	Port pin as input		
6	Software delay		
7	Bit manipulation		
8	Usage of for loop		
9	16 x 2 LCD Display		
10	7 Segment LED Display		
11	ADC Interfacing		
12	External Interrupt		
13	Stepper motor interfacing		
14	Study of Timer		
15	UART		
16	PROJECT 1 Temperature Controller		
17	PROJECT 2 Bluetooth Controller		
18			
General Remarks: (For office use only)			
Test 1:		Test 2:	
Assign 1:		Assign 2	

VISION AND MISSION

Government Polytechnic College, Perumbavoor Vision and Mission

Vision

Excel as a centre of skill education moulding professionals who sincerely strive for the betterment of society.

Mission

- To impart state of the art knowledge and skill to the graduate and moulding them to be competent, committed and responsible for the well being of society.
- To apply technology in the traditional skills, thereby enhancing the living standard of the community

Department of Electronics & Communication Engineering

Vision

Excel as a centre of skill education in Electronics and Communication Engineering, moulding professionals who sincerely strive for the betterment of themselves and society.

Mission

- To impart state of the art knowledge, skill and attitude to the students and contributing to their sustainable development.
- To merge technologies in the field of Electronics and Communication Engineering with occupational skills, thereby improving quality of living.

Exp No.

1

Date:

D	D	-	M	M	-	Y	Y
---	---	---	---	---	---	---	---

PEO, PO and PSOs of the Program

Program Educational Outcome (PEOs)

PEO1: Secure successful careers in manufacturing, testing, maintenance, development and marketing in Electronics and Communication Engineering.

PEO2: Acquire knowledge and competency in the domain to develop innovative, cost effective and socially acceptable solutions to engineering problems in a multi disciplinary work environment.

PEO3: Develop strong fundamental knowledge that prepares them for professional careers/higher studies with attitude for lifelong learning.

PEO4: Instill the attitude to be sensitive to ethical, societal and environmental issues while pursuing their professional duties.

PEO5: Possess leadership qualities and be effective communicator to work efficiently with diverse teams, promote and practice appropriate ethical practices.

Program Outcomes (POs)

PO1: Basic Knowledge: Apply knowledge of basic mathematics, science and engineering to the solution of engineering problems.

PO2: Discipline knowledge: Apply knowledge in Electronics and Communication Engineering to solve applied engineering problems.

PO3: Experiments and practice: Able to plan and perform experiments and practices and to solve engineering problems.

PO4: Engineering Tools: Apply appropriate technologies and tools with proper understanding of their limitations.

PO5: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO6: Environment and sustainability: Understand the impact of the engineering solutions in societal and environmental contexts and demonstrate the knowledge and need for sustainable development.

PO7: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO8: Individual and team work: Function effectively as an individual and as a member or leader in diverse multidisciplinary teams.

PO9: Communication: Able to communicate effectively.

PO10: Lifelong learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the context of technological changes.

Program Specific Outcome (PSO):

PSO1: Specialization knowledge: Apply concepts and knowledge in the field of semiconductor devices, communication and networking technologies, embedded systems.

PSO2: Professional growth: Generate ideas from the knowledge of engineering specialization leading to professional growth.

PSO3: Entrepreneurship: Apply knowledge and understanding of engineering principles to initiate entrepreneurship ventures.

Exp No.

Date:

SAFETY PROCEDURES

Problem Statement:

The safety instructions are presented to the attention of the students as a mean of preventing accidents while performing experiments and activities in the communication lab of the department .The purpose is to draw attention to the risks involved in lab activities to prevent human suffering and damage to equipment.

Safety in the laboratory:

Working in the lab is not allowed without following electricity precautions displayed.

No individual work is allowed in the lab.

Laboratory in charge is responsible for the arrangements of your lab activities; Listen carefully to his/her instructions and follow them.

To do and not to do:

Inform the lab in charge about dangerous conditions and faults in the lab or nearby environment.

Do not do any action that may harm people or equipments in the lab.

Do not misuse any of the tools or instruments belong to the lab.

Strict discipline should be maintained in the laboratory.

Turn off cell phones before entering the lab.

At the end and beginning of laboratory, follow 5S procedures and leave the work table clean and tidy.

Electrical Safety:

Consult Electrical Engineering section available in the campus for electrical safety queries.

The lab equipment is powered from electrical sockets installed on the tables. Do not use equipment that is powered from a damaged socket.

Do not use equipment that is powered from flexible cable with damaged insulation or if it's plug is not assembled properly.

Do not repair or disassemble electrical equipment including replacement of fuses installed in the equipment.

Do not open the main fuse box, unless it is an emergency and you need to switch off main circuit breaker.

Emergency Switches:

The laboratory has circuit breakers, which is located in the main panel. Identify the place. In an emergency condition, switch off circuit breakers immediately.

Result

Familiarization of safety precautions performed.

	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No.

Date:

HANDLING ELECTROSTATIC DISCHARGE (ESD)

Problem Statement:

Familiarize ESD handling procedures in the laboratory

Theory

In handling electronic devices, datasheets cautions about ESD (Electrostatic Discharge) precautions. These devices are prone to damage because of electrostatic charges made by human body. These charges may be up to 4000 volts and cause damage without being noticed. It is recommended to follow ESD precautions on handling of these devices.

Points for the elimination of ESD damage to electronic components

1. Make sure you have a reliable ground point available near the table.
2. Do not wear clothing which generates static electric charges every time you move.
3. Do not handle static generating objects while working on electronics.
4. Store all chips and other components in appropriate anti-static containers.
5. Keep all ESD sensitive components and spares in anti-static envelopes for storage.
6. Be sure to turn off the power and remove the power plug from all equipment before working repairing or assembling.
7. Do not plug in or remove equipments while the power is on.

Result

Familiarization of ESD protection procedures performed.

	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No.

4

Date:

D	D	-	M	M	-	Y	Y
---	---	---	---	---	---	---	---

FAMILIARIZATION OF AVR STUDIO 6

Problem Statement : To familiarize AVR Studio6 for ASM and C coding of ATMEGA 32 and programming software (extreme burner)

Theory

Atmel AVR studio 6 is an integrated development environment by Atmel. It provides a modern powerful and simple environment to write programs in Assembly, C and C++ language and also helps in debugging programs in windows XP, windows VISTA and windows 7/8 environments. It supports all AVR ICs by Atmel and also new AVR/ARM devices. It provides the same platform for 8-bit, 32-bit and ARM series of microcontroller. Atmel Studio 6 has its own C compiler that's why there is no need for external C compiler.

Steps for using AVR Studio

- Step-1: Download ATMEL studio 6
- Step-2: Open atmel studio
- Step-3: Create a new project
- Step-4: Device selection
- Step-5: Write a program
- Step-6: Compile the code
- Step-7: Open programmer software (extreme burner)
- Step-8: Select hex file generated
- Step-8: Press write hex menu

Result

Familiarization of AVR studio and Extreme burner performed.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

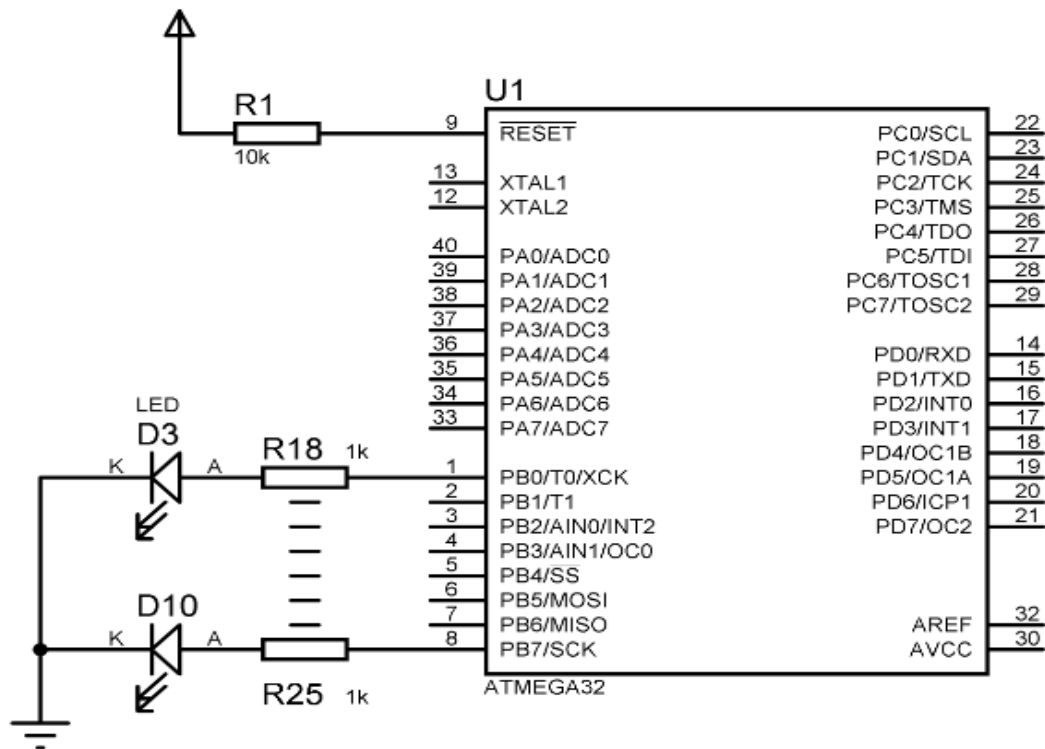
Exp No. 5

Date: D D - M M - Y Y

STUDY OF PORTS

Problem Statement : To send one byte of data on to PORTB of ATMEGA 32 there by understanding the operation of PORT as output.

Circuit Diagram



Theory

ATMEGA ports are 8 bit wide. Each port has 3 eight bit registers associated. Each bit in these registers configures pins of associated port. Bit 0 of these registers is associated with Pin 0 of the port, Bit1 of these registers is associated with Pin1 and so on.

These three registers are

- DDRx register
- PORTx register
- PINx register

x may be replaced by A,B,C or D based on the PORT you are using.

DDRx register

DDRx (Data Direction Register) configures data direction of the port pins. Which, writing 0 to a bit in DDRx makes corresponding port pin as input, while writing 1 to a bit in DDRx makes the corresponding port pin as output.

example:

- to make all pins of port B as input,
DDRA = 0b00000000;
- to make all pins of port A as output pins :
DDRB= 0b11111111;
- to make lower [nibble](#) of port B as output and higher [nibble](#) as input :
DDRB = 0b00001111; In hexadecimal representation, it can be written as DDRB = 0x0F;

Algorithm

1. Set PORTB as output by suitably initializing DDRB register
2. Move required value to PORTB register
3. Go to indefinite loop.

Code

<i>Write general comments about the program here</i>		
This program demonstrates the working of port Author : XXXXXXXX Date : XX.XX.20XX Program set LED connected to PORTB alternately on PB7 -1 PB6-0 PB5-1 PB4-0 PB3 -1 PB4-0 PB1-1 PB0-0 -----*/		
Label	Instructions	Comments

--	--	--

Result

One byte of data moved to PORTB and output verified on LED connected to PORTB.

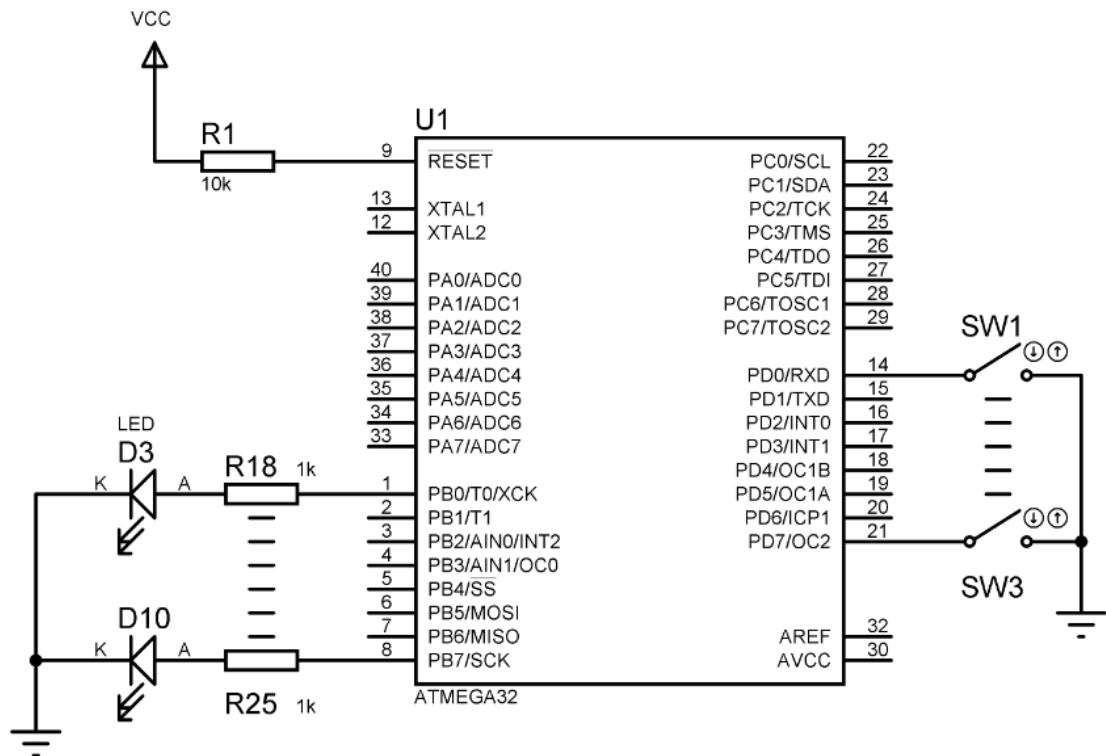
Value sent to PORTB in hex =

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

PORT PIN AS INPUT

Problem Statement : To read switches connected to PORTD and output the contents to LEDs connected to PORTB.

Circuit Diagram



Theory

ATMEGA ports are 8 bit wide. Each port has 3 eight bit registers associated. Each bit in these registers configures pins of associated port. Bit 0 of these registers is associated with Pin 0 of the port, Bit1 of these registers is associated with Pin1 and so on.

These three registers are

- DDRx register
- PORTx register
- PINx register

x may be replaced by A,B,C or D based on the PORT you are using.

DDRx register

DDRx (Data Direction Register) configures data direction of the port pins. Which, writing 0 to a bit in DDRx makes corresponding port pin as input, while writing 1 to a bit in DDRx makes the corresponding port pin as output.

example:

- to make all pins of port B as input,
DDRA = 0b00000000;
- to make all pins of port A as output pins :
DDRB= 0b11111111;
- to make lower [nibble](#) of port B as output and higher [nibble](#) as input :
DDRB = 0b00001111; In hexadecimal representation, it can be written as DDRB = 0x0F;

Algorithm

1. Set PORTB as output by suitably initializing DDRB register
2. Set PORTD as output by suitably initializing DDRD register
3. Enable internal pull ups by writing 0xff to PORTB
4. Read value of PIND
5. Write the value to PORTD
6. Go to step 4

Code

Write general comments about the program here

```
/*-----  
  
This program demonstrates the working of ports as input and output  
Author: xxxxxxxxx Date: xx.xx.20xx  
Program reads switches connected to PORTD  
and output the values to PORTB  
-----*/
```

Label	Instructions	Comments

--	--	--

Result

One byte of data moved to PORTB and output verified on LED connected to PORTB.

Value sent to PORTB in hex =

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 7

Date: D D - M M - Y Y

SOFTWARE DELAY

Problem Statement : To blink LEDs connected to PORTB at 1 sec interval while buzzer and relay connected to PORTC beeps.

Circuit Diagram (refer to experiment 4)

Algorithm

1. Ensure proper setting of fuse bits to enable PORTC
2. Set PORTB as output by suitably initializing DDRB register
3. Set PORTC as input by suitably initializing DDRC register
4. Set PORTB and PORTC high
5. Invert PORTB
6. Invert PORTC

Set value of fuse bit as Fuse high = C9 and fuse low = E1 for enabling PORTC and clock at internal RC 1 MHz

Code

Write general comments about the program here

```
/*-----  
  
This program blink LED connected to PORTB ON OFF  
  
and relay and buzzer connected to PORTC at 1sec interval  
with internal RC oscillator at 1Mhz  
author: xxxxxxxx      Date: xx.xx.20xx  
Ver: 1.0      Modifications made: NIL  
-----*/
```

Label	Instructions	Comments

--	--	--

Result

One byte of data moved to PORTB and output verified on LED connected to POERTB.

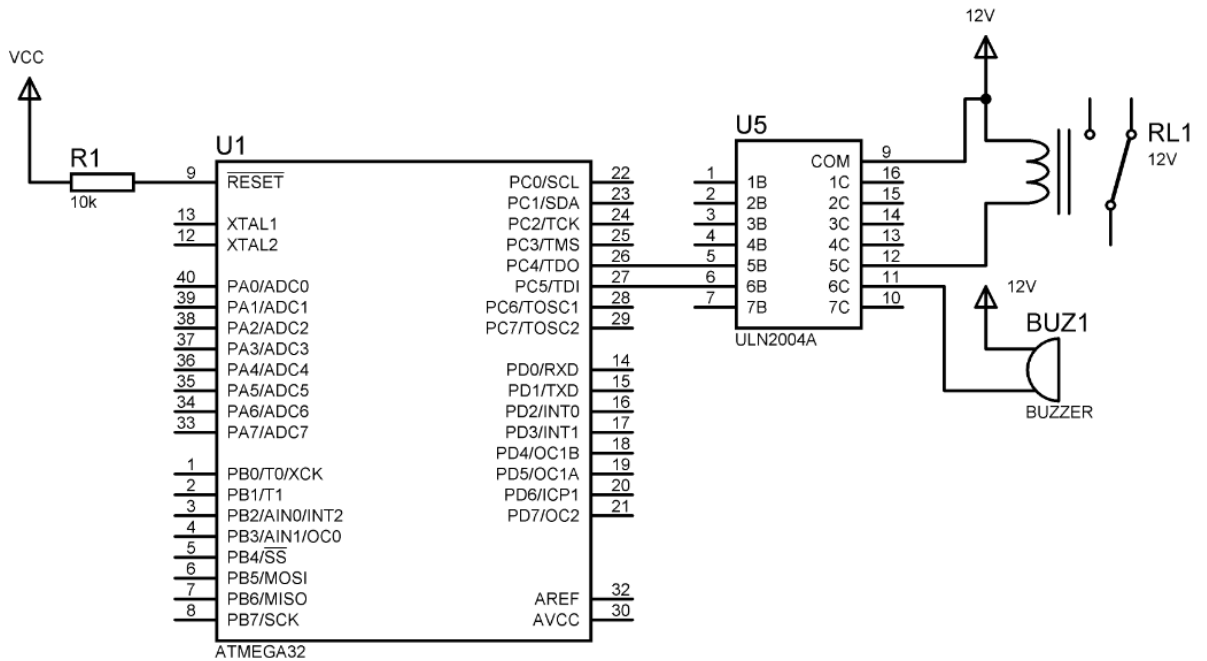
For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 8

Date: D D - M M - Y Y

BIT MANIPULATION

Problem Statement : To blink LEDs connected to PORTB and buzzer connected to PC5 at 1 sec interval relay connected to PC4 is on using logical operations.



Algorithm

1. Ensure proper sitting of fuse bits to enable PORTC
2. Set PORTB as output by suitably initializing DDRB register
3. Set PORTC as input by suitably initializing DDRC register
4. Set PORTB and PORTC high
5. Invert PORTB
6. Invert PORTC

Set value of fuse bit as Fuse high = C9 and fuse low = E1 for enabling PORTC and clock at internal RC 1 MHz

Code

Write general comments about the program here

```
/*-----  
This program blink LEDs connected to PORTB and buzzer connected to PC5 at 1 sec  
interval relay connected to PC4 is on using logical operations.  
The program assumes internal RC oscillator at 1Mhz  
Author : xxxxxxxxxxxxxxxx      Date   : xx.xx.20xx  
-----*/
```

Label	Instructions	Comments

Result

Performed blinking of LEDs connected to PORTB and buzzer connected to PC5 at 1 sec interval relay connected to PC4 is on using logical operations.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No.

Date:

USAGE OF FOR LOOP

Problem Statement : Write a program in C using for loop to blink LED connected to PORTB at 1sec delay 100 times

Algorithm

1. Set PORTB as output by suitably initializing DDRB register
2. Set PORTB and PORTC high
3. Use for loop to increment 10 times and send value to PORTB
4. Go to indefinite loop

Code

```
Write general comments about the program here

/*-----
This is a program to blink LED connected to PORTB at 1 sec interval 10 times
Author: xxxxxxxxxxxx      Date: xx.xx.20xx
Ver: 1.0      Modifications made: NIL
-----*/
```

Label	Instructions	Comments

--	--	--

Result

Performed blinking of LEDs connected to PORTB and buzzer connected to PC5 at 1 sec interval relay connected to PC4 is on using logical operations.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 10

Date: DD-MM-YY

16 x 2 LCD Display

Problem Statement : Given lcd.h, write a program in C to display your name in first row and class number in second row of 16 x 2 LCD display in 8 bit mode.

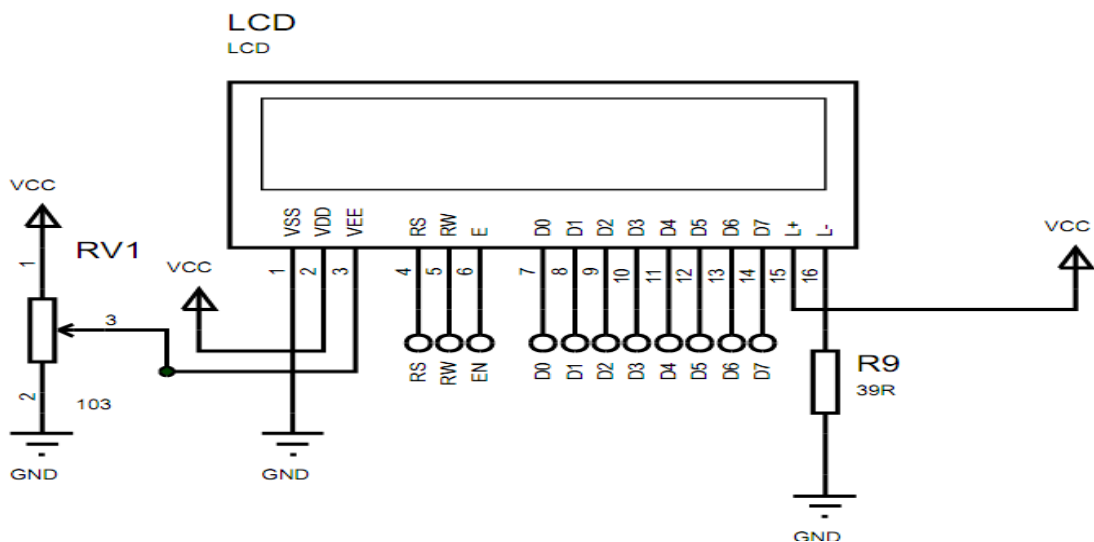
D0 to D7 connected PB0 to PB7 to RS – PD4, RW – PD5, EN – PD6 (keep RW to GND)

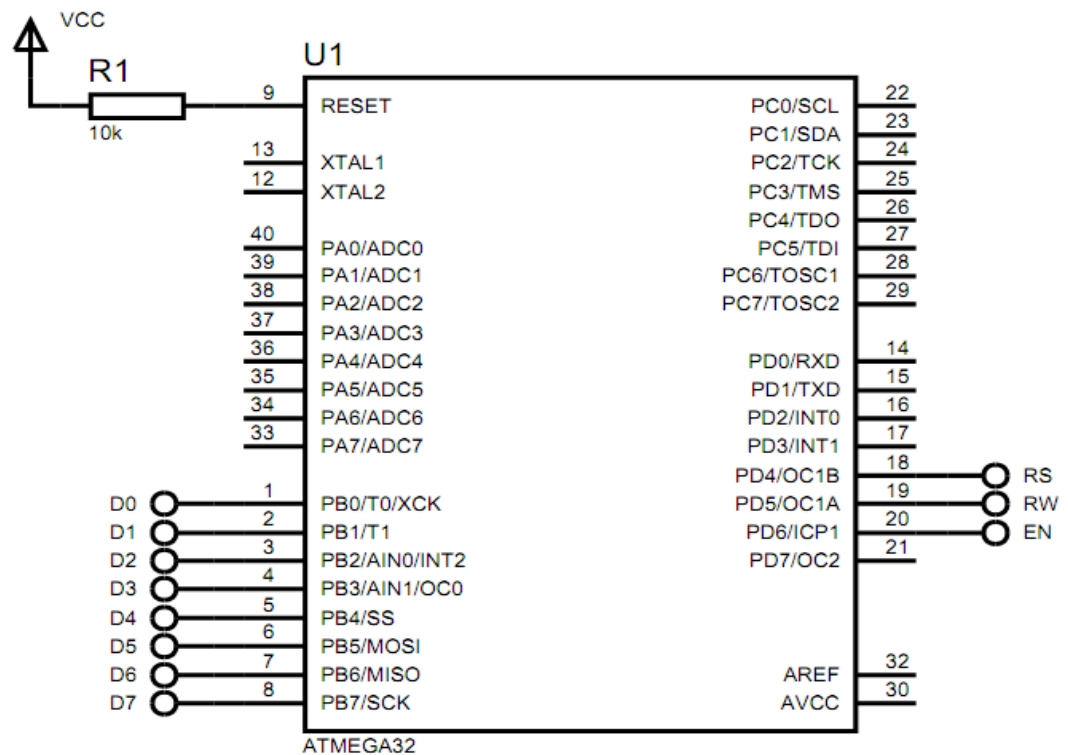
Information given

```
#define D0 eS_PORTB0
#define D1 eS_PORTB1
#define D2 eS_PORTB2
#define D3 eS_PORTB3
#define D4 eS_PORTB4
#define D5 eS_PORTB5
#define D6 eS_PORTB6
#define D7 eS_PORTB7
#define RS eS_PORTD4
#define EN eS_PORTD6

Lcd8_Init(); // LCD initialization
Lcd8_Set_Cursor(1,1); // set cursor first line first column
Lcd8_Write_String("LINE 1"); // String send to display
```

Theory





LCD display is an inevitable part in almost all embedded projects. It consists of 16 rows and 2 columns of 5×7 or 5×8 LCD dot matrices. They are available in a 16 pin package with back light, contrast adjustment. The pin numbers, their name and corresponding functions are shown above.

LCD Initialization.

The steps that have to initializing the LCD display is given below.

- Send 38H to the 8 bit data line for initialization
- Send 0FH for making LCD ON, cursor ON and cursor blinking ON.
- Send 06H for incrementing cursor position.
- Send 01H for clearing the display and return the cursor.

Sending data to the LCD.

The steps for sending data to the LCD module are given below. Make R/W low.

- Make RS=0 if data byte is a command and make RS=1 if the data byte is a data to be displayed.
- Place data byte on the data register.
- Pulse E from high to low.
- Repeat above steps for sending another data.

Algorithm

1. Include lcd.h file
2. Initialize PORTB and D
3. Initialize LCD
4. Set cursor to row0 column 1
5. Sent string to be displayed in row 0
6. Set cursor to row1 column 1
7. Sent string to be displayed in row 1
8. Go to indefinite loop.

Code

<i>Write general comments about the program here</i>		
<pre>/*----- This is a program display name in first line and sl no in second line of 16x2 LCD. D0 10 D7 connected to PB0 to 7, RS - PD4, RW - PD5, En - PD6 Author: xxxxxxxxxxxxxx Date: xx.xx.20xx Ver: 1.0 Modifications made: NIL -----*/</pre>		
Label	Instructions	Comments

--	--	--

Result

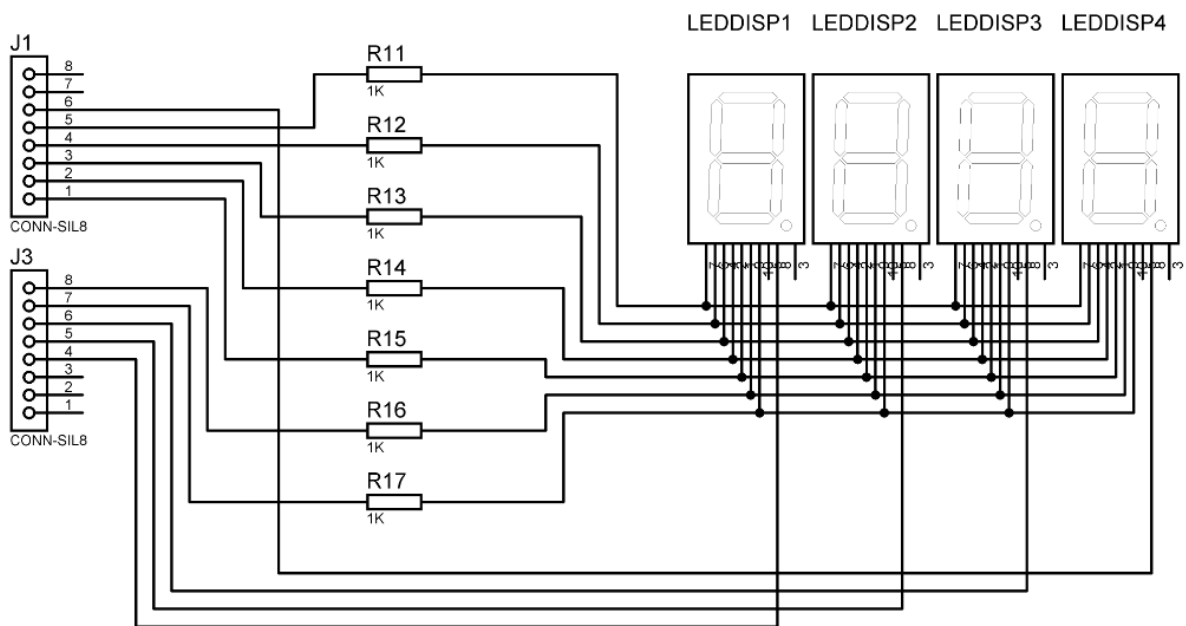
Values 0x00 0xFF displayed on PORTB at 1sec delay

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

7 Segment LED Display

Problem Statement : To display '0123' on multiplexed seven segment cathode display with segments connected to PB0 to PB6 and digits connected to PD4, PD5, PD6 and PB7 respectively.

Theory



Seven segment displays are available in common anode and common cathode configuration. They are usually used in multiplexed configuration. One segment from each of the digits is connected to a common line which is controlled by a single output pin from the microcontroller. One digit is activated at a time in this method.

Algorithm

1. Initialize ports
2. Send code corresponding to zero to segments
3. Select digit by making digit select low
4. Delay for 10 msec
5. Disable segments
6. Send value corresponding to digit 1
7. Repeat steps 2 to 5 for all four digits
8. Go to step2

--	--	--

Result

Values "0123" displayed on common cathode display

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No.

Date:

ADC Interfacing

Problem Statement : To read temperature sensor connected to AD0 pin of ATMEGA32 at 10 bit resolution and transfer the content in ASCII format to PORTB. Bits should be sent at an interval of 2 seconds.

Information provided

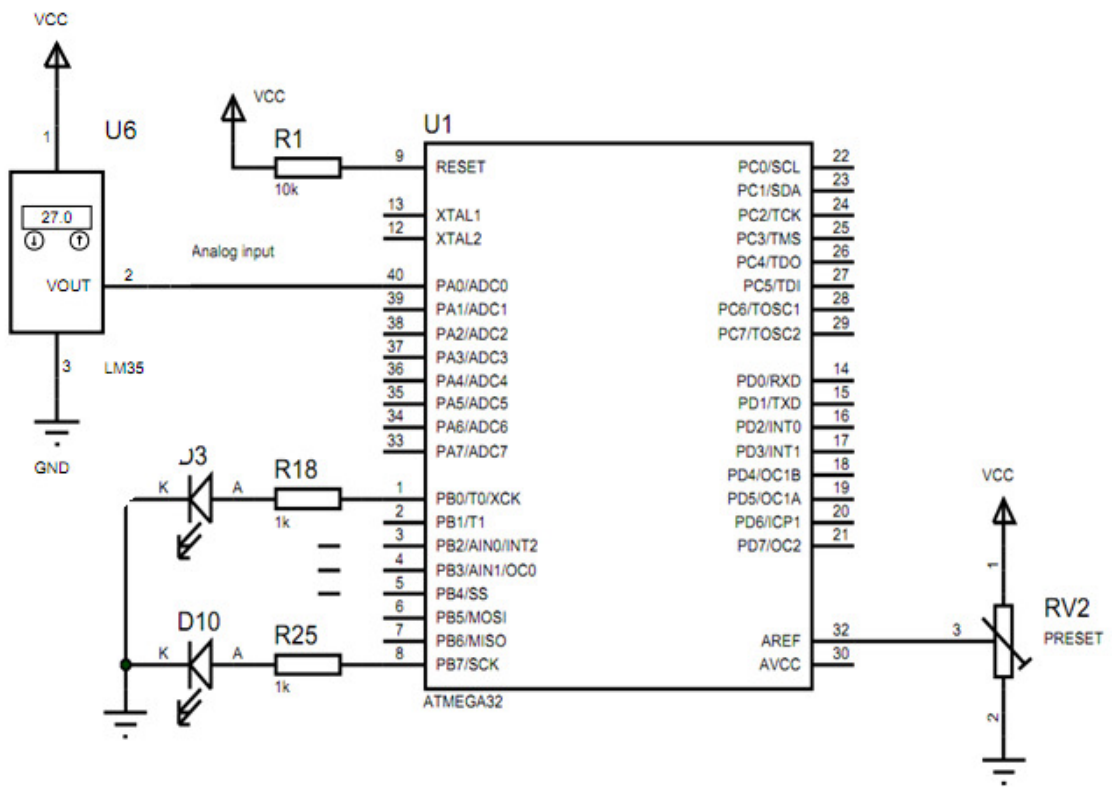
```
// ----- ADC initialization -----  
void InitADC()  
{  
    ADMUX &= ~(1<<REFS0);           // For Aref=EXTERNAL  
    ADMUX &= ~(1<<REFS1);           // For Aref=EXTERNAL  
    ADCSRA=(1<<ADEN)|(7<<ADPS0);    // ADC enable, Clock/2  
}  
  
// ----- read adc -----  
uint16_t ReadADC(uint8_t ch)  
{  
    ch=ch&0b0000111;                //Select ADC Channel ch must be 0-7  
    ADMUX|=ch;                       //Select ADC Channel  
    ADCSRA|=(1<<ADSC);               //Start Single conversion  
    while(!(ADCSRA & (1<<ADIF)));    //Wait for conversion to complete  
    ADCSRA|=(1<<ADIF);               //Clear ADIF by writing one to it  
    return(ADC);                     // return ACD values  
}
```

Theory

The ADC is multiplexed with PORTA.

ADC Prescaler: The ADC needs a clock pulse to do its conversion. This clock generated by system clock by dividing it to get smaller frequency. At higher frequency the conversion is fast while a lower frequency the conversion is more accurate. The prescaler is provided to produces acceptable frequency for ADC from any system clock frequency. System clock can be divided by 2,4,16,32,64,128 by setting the Prescaler.

ADC Channels: The ADC in ATmega32 has 8 channels .We can connect up to 8 different sensors and get their values separately.



Algorithm

1. Initialize ports
2. Initialize ADC
3. Read ADC
4. Convert to ascii string
5. Send to PORTB at 2 sec
6. Repeat 3 times from 4
7. Call 3 sec delay
8. Clear led
9. Go to step3

Code

Write general comments about the program here

```
/* To read temperature sensor connected to AD0 pin of ATMEGA32 and transfer
 * the content in ASCII format to PORTB.
 * Least significant 2 bits should be sent at an interval of
 * 2 seconds after the most significant byte.
 * adc.c
 * Author: XXXXXXXXXXXXX Created: XX/XX/20XX
-----*/
```

Label	Instructions	Comments

--	--	--

Result

Values of temperature sensor displayed as string to PORTB

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

External Interrupt

Problem Statement : To toggles BUZZER connected to PC5 when interrupted with INT0 and relay connected to PC4 when interrupted with INT1.

Information provided

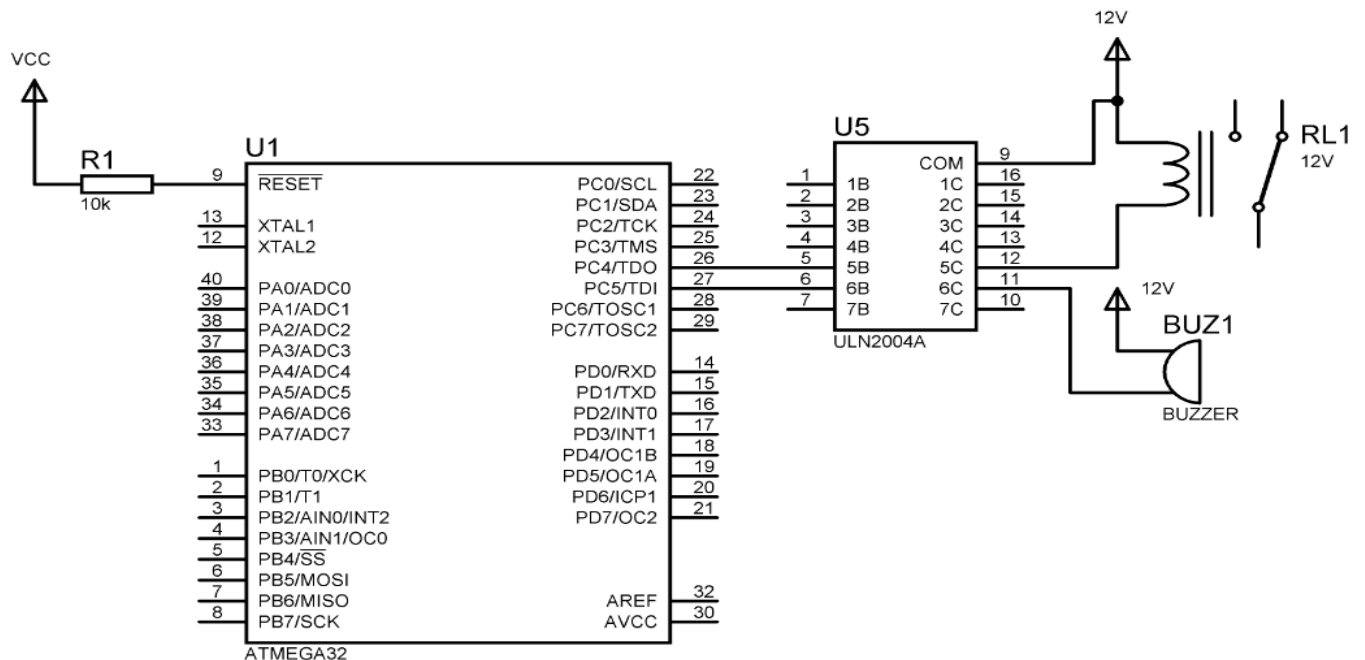
General Interrupt Control Register – GICR

Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	GICR
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

MCU Control and Status Register – MCUCSR

Bit	7	6	5	4	3	2	1	0	
	JTD	ISC2	–	JTRF	WDRF	BORF	EXTRF	PORF	MCUCSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0						See Bit Description

Theory



The External Interrupts are triggered by the INT0, INT1, and INT2 pins. Observe that, if enabled, the interrupts will trigger even if the INT0..2 pins are configured as outputs. The external interrupts can be triggered by a falling or rising edge or a low level (INT2 is only an edge triggered interrupt). This is set up as indicated in the specification for the MCU Control Register – MCUCR – and MCU Control and Status Register – MCUCSR. When the external interrupt is enabled and is configured as level triggered (only INT0/INT1), the interrupt will trigger as long as the pin is held low

Algorithm

1. Initialize ports
2. Initialize interrupt
3. Indefinite loop

ISR INT0
Toggle relay

ISR INT1
Toggle buzzer

CODE

<i>Write general comments about the program here</i>		
<pre> /* Program to demonstrate external interrupt0 and 1. toggles BUZZER connected to PC5 when interrupted with INT0 toggles RELAY connected to PC4 when interrupted with INT1 Author: XXXXXXXXXXXXXXXX Created: XX/XX/20XX -----*/ </pre>		
Label	Instructions	Comments

--	--	--

Result

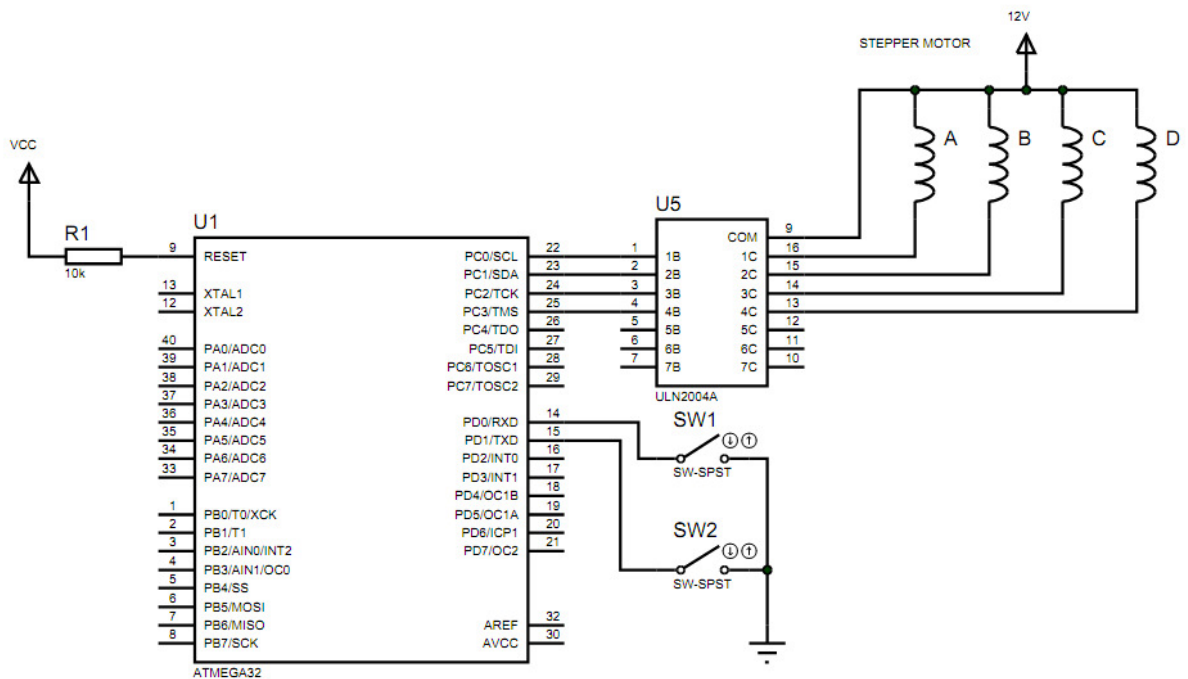
Buzzer connected to PC5 when interrupted with INT0 and relay connected to PC4 when interrupted with INT1.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Stepper motor interfacing

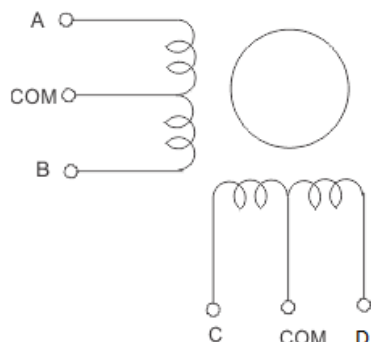
Problem Statement : To interface stepper motor to port pins PC3, PC2, PC1 and PC0. To control forward and reverse motion by switches connected to PD0 and PD1 respectively.

Theory



Theory:

A stepper motor is a type of DC motor that rotates in steps. When digital sequence is applied to it, the motor rotates in steps and the speed of rotation depends on the rate at which the sequences are applied and the direction of rotation is dependent on the pattern of pulses that is followed.



The stepper motor is made up of a rotor, which is normally a permanent magnet and it is, as the name suggests the rotating component of the motor. A stator is another part which is in the form of winding.

Step	A	B	C	D
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0
4	0	0	1	1

In the diagram below, the center is the rotor which is surrounded by the stator

Algorithm

1. Initialize ports
2. Check for key press
3. If forward go to 6
4. If reverse go to 7
5. Wait for key press
6. Send forward sequence with delay
7. Send reverse sequence with delay
8. Goto 2

Code

Write general comments about the program here

```

/* Program to interface stepper motor to port pins PC3, PC2, PC1 and
PC0. To control forward and reverse motion by switches connected to
PD0 and PD1 respectively.
Author: XXXXXXXXXXXXXXXX Created: XX/XX/20XX
-----*/
-----*/

```

Label	Instructions	Comments

--	--	--

Result

Stepper motor interfaced with forward and reverse rotation.

.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No. 15

Date: D D - M M - Y Y

Study of Timer

Problem Statement : To generate square wave at PORTB using timer in interrupt mode.

Theory

Timers generally have a resolution of 8 or 16 Bits. An 8 bit timer is 8 bit wide so capable of holding value within 0-255. But these register increment or decrement automatically at a predefined rate based on the timer clock. This operation does not need CPU's intervention. That is its counts up to its maximum value (255 for 8 BIT timers) and rolled back to 0. In this situation timer can issue an interrupt and you must write an Interrupt Service Routine (ISR) to handle the event.

Using 8 BIT Timer (TIMER0)

The ATmega32 has three different timers of which the simplest is TIMER0. Its resolution is 8 BIT i.e. it can count from 0 to 255. The Prescaler is a mechanism for generating clock for timer by the CPU clock. As you know that CPU has a clock source such as a external crystal of internal oscillator. Normally these have the frequency like 1 MHz, 8 MHz, 12 MHz or 16MHz(MAX). The Prescaler is used to divide this clock frequency and produce a clock for TIMER.

Algorithm

1. Initialize ports.
2. Initialize timer interrupt.
3. Indefinite loop.

ISR Timer 0

1. Increment counter.
2. If counter value greater than preset toggle PORTB.
3. Else increment count.

Code

Write general comments about the program here

```
/* This program demonstrate working of timer in interrupt mode to generate square wave at PORTB
```

```
Author: XXXXXXXXXXXXXXXX Created: XX/XX/20XX
```

```
-----*/
```

Label	Instructions	Comments

--	--	--

Result

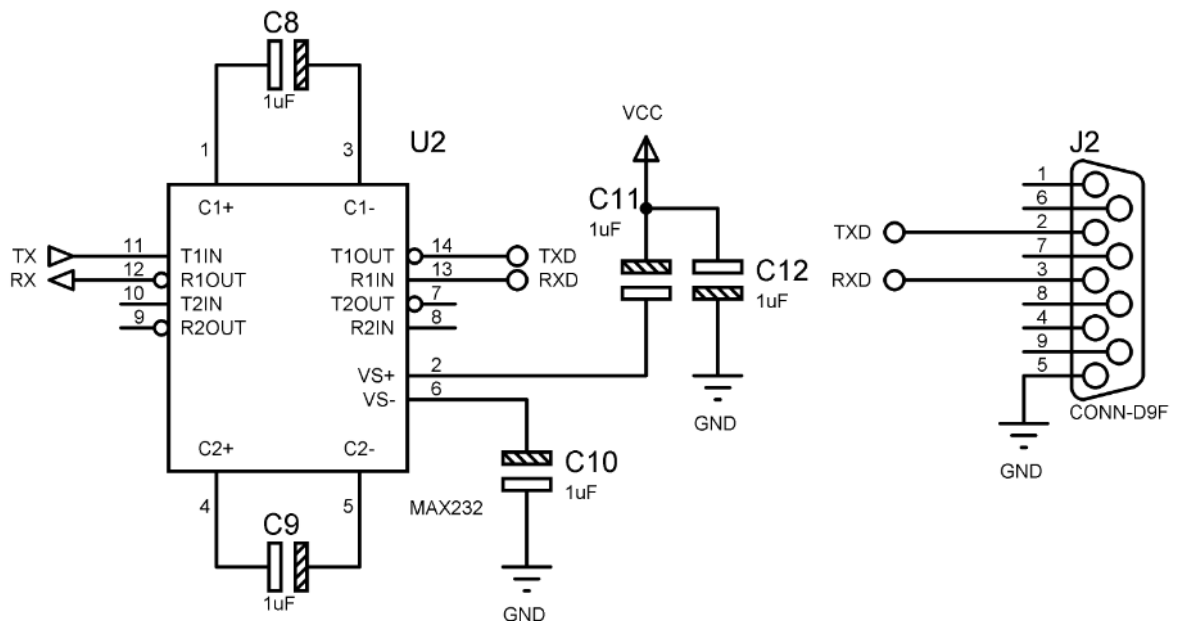
Square wave generated using Timer 0.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

UART

Problem Statement : Write a program in C to sent ASCII value 'A' to USART at 9600 baud rate. Any value received from USART is reverted back instead of 'A' and displayed on PORTB with external crystal oscillator at 11.0592 Mhz

Theory



Algorithm

1. The UCSRB register is loaded with the value 08H, enabling the USART transmitter.
The transmitter will override normal port operation for the TxD pin when enabled.
2. The UCSRC register is loaded with the value 06H, indicating asynchronous mode with 8-bit data frame, no parity, and one stop bit.
3. The UBRR is loaded with one of the values in Table 11-4 (if $F_{osc} = 8 \text{ MHz}$) to set the baud rate for serial data transfer.
4. The character byte to be transmitted serially is written into the UDR register.
5. Monitor the UDRE bit of the UCSRA register to make sure UDR is ready for the next byte.
6. To transmit the next character, go to Step 4.

Code

Write general comments about the program here

```
/* This program sent ASCII value 'A' to USART at 9600 baud rate. The value is also  
send to PORTB  
Any value received from USART is reverted back instead of 'A' and displayed on  
PORTB with internal RC oscillator at 1Mhz  
Author: XXXXXXXXXXXXX Created: XX/XX/20XX Ver: 1.0 Modifications made: NIL  
-----*/
```

Label	Instructions	Comments

--	--	--

Result

UART programmed at 9600 baud rate to send and receive values.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No.

Date:

PROJECT 1 Temperature Controller

Problem Statement : Control Buzzer/ Relay connected to PORTC based on temperature reading through LM 35 connected to ADO. Display the value on LCD display.

Code

Write general comments about the program here

```
/* this project read temperature sensor LM35
 * and display on LCD
 * temp_disp.c
 Author: XXXXXXXXXXXX Created: XX/XX/20XX
-----*/
```

Label	Instructions	Comments

--	--	--

Result

Project implemented to display value of temperature sensor on LCD display and control buzzer accordingly.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No.

Date:

PROJECT 2 Bluetooth Controller

Problem Statement : Control relay and buzzer connected to PORTC using Bluetooth.

Code

Write general comments about the program here

```
/* this project Control relay and buzzer connected to PORTC using Bluetooth
 * temp_disp.c
 * Author: XXXXXXXXXXXXX Created: XX/XX/20XX
-----*/
```

Label	Instructions	Comments

--	--	--

Result

Bluetooth controller implemented.

For Office use only	Signature of Lab in charge	Remarks
Readiness to do experiment		
Completion of Experiment		

Exp No.

Date: --

Exp No.

Date:

D	D	-	M	M	-	Y	Y
---	---	---	---	---	---	---	---

Exp No.

Date:

D	D	-	M	M	-	Y	Y
---	---	---	---	---	---	---	---

Exp No.

Date:

D	D	-	M	M	-	Y	Y
---	---	---	---	---	---	---	---



GOVERNMENT POLYTECHNIC COLLEGE, PERUMBAVOOR

Koovappady P O, Perumbavoor, Ernakulam – 683544

www.gptcperumbavoor.ac.in
