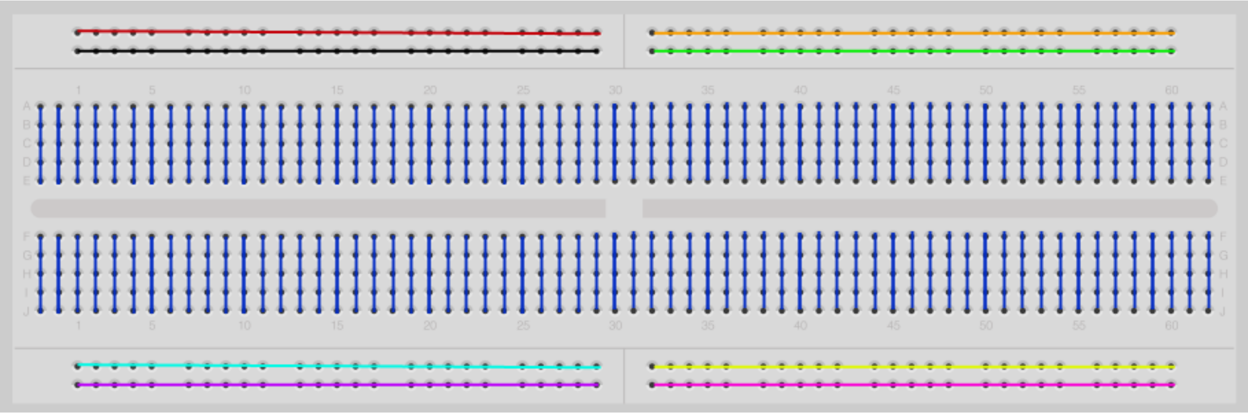
KiDOBOTiKZ

Engineering Real Engineers

INTERMEDIATE LEVEL NOTES

(LEVEL 4, 5 and 6)

UNDERSTANDING YOUR BREAD BOARD



Breadboard is a prototyping tool for making an experimental model of a circuit. It consists of numerous holes (for inserting the components) and these holes are internally connected in different patterns as shown in the picture

Multimeter is an instrument used to measure various electrical and electronic parameters such as voltage, current, resistance etc.

The continuity test in a multimeter helps to check whether a connection exists between two points. This functions by passing few milliamps current through the red terminal and a buzzer is connected to the black terminal.

ELECTRONIC COMPONENTS

RESISTOR

How it works?

Resistor consists of a small ceramic (clay) tube covered partially by a conducting carbon film. The composition of the carbon determines how much current can pass through. When two resistors are connected in series, the resistance increases.

Uses:

The resistor causes drop in the voltage and hence can be used to reduce the voltage for needs such as glowing an LED with 5V input (220Ω resistor should be added)

Polarity identification:

No polarity

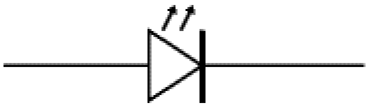
Unit/Part number:

Ohms (Ω). No Part-number but to identify the value colour codes/numbers are given Black – 0 Brown – 1 Red – 2

Orange – 3 Yellow – 4 Green – 5 Blue – 6

Violet – 7 Green – 8 White – 9

LED



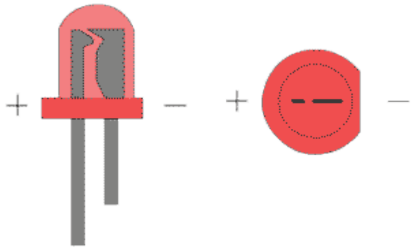
How it works?

LED (Light Emitting Diode) is a type of diode wherein the interaction between the positive and negative charges causes emission of light.

Uses:

For indication purposes

Polarity identification:



Long leg – +ve

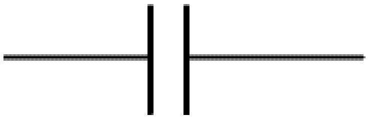
Short leg/Flat side – -ve

Unit/Part number:

-NA- (They are only identified by the size, colour and power)

Date:

CAPACITOR



How it works?

Capacitor is an electronic component that stores electric energy (a battery produces electrons).

Uses:

When connected in parallel, the capacitor dissipates the stored charges to the circuit during supply voltage fluctuations and thus protects the circuit.

When connected in series, it allows current only till it gets charged to 3/4th its value (only few seconds) and then becomes open circuit.

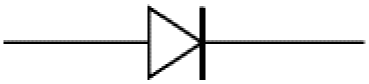
Polarity identification:

Long leg - +ve, Short Leg - -ve (it will also be marked on the casing of the capacitor

Unit/Part number:

The charge of a capacitor is measured in units called Farad (F).

DIODE



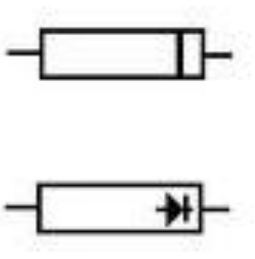
How it works?

A diode consists of a semi-conducting material with Positive and negative regions joined by a junction. Initially, at the junction, the positive and negative charges combine and form a blocking region. When + is given to the +ve region, they start repelling and the +ve particles are forced to move through the blocking region, hence current starts flowing. But if –(minus) is given to +ve region, they both attract each other and current does not flow.

Uses:

As it allows the current to pass through only in one direction, it can be used for reverse polarity protection

Polarity identification:



Unit/Part number: IN4007 is the part number used

Date:

VOLTAGE REGULATOR IC

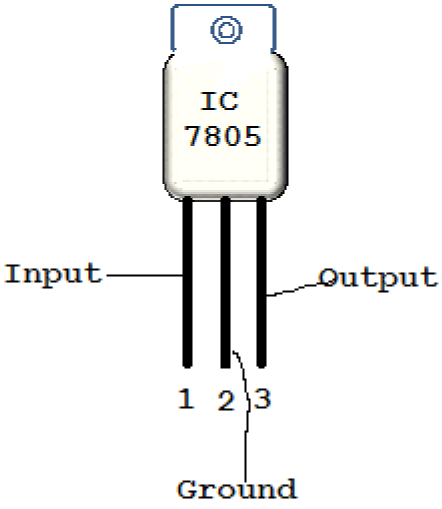
What is its function?

Its function is to automatically maintain a constant output voltage level, which is lesser than the input voltage. Two types - +ve voltage regulator (78xx) and –ve voltage regulator (79xx). xx-output voltage

Uses:

7805 is used when the supply voltage is 12v or above (required for motors etc) but the electronic circuit needs 5V for its functioning.

Pinout diagram:



COMPARATOR IC

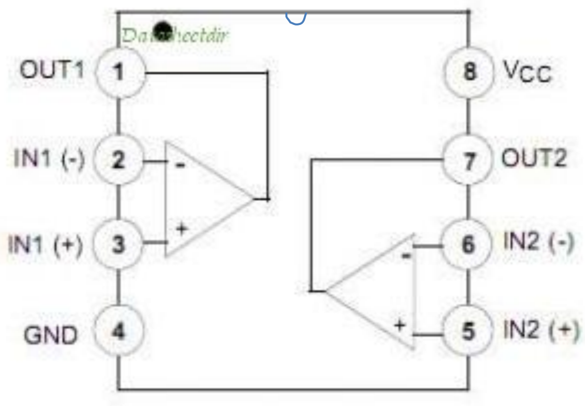
What is its function?

A comparator's function is to compare two analog input voltages and give a digital output voltage. If IN+ is greater the IN–, then the output voltage will be 5V. If IN– is greater than IN+, then output voltage will be 0V.

Uses:

Is used in most of the digital sensors to convert the analog values given by the sensing element into a digital output

Pinout diagram:



LM258

DUAL OPAMP IC

Date:

RELAY

What is its function?

A relay is an electromechanical switch that get activated when using a small voltage signal. It consists of a coil and a metal flap (which gets attracted towards the coil, when current is passed through the coil).

Uses:

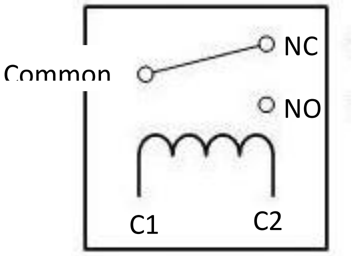
Switch high voltage/current DC and AC loads using low voltage DC signal

Pinout diagram:

NC – Normally closed

NO – Normally open

C1, C2 – coil ends



Date:

TRANSISTOR

What is its function?

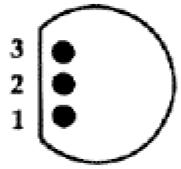
Transistor is a semiconductor that can function as a switch. It consists of three terminals (base, emitter and collector). When the base is triggered, the collector and emitter gets connected. A transistor is much smaller than a relay and can handle much smaller loads only (only DC). But since there are no moving parts as in a relay, transistors are more reliable.

Uses:

Switch high voltage DC loads using low voltage DC signal

Pinout diagram:

2N2222N transistor



1 – Emitter

2- Base

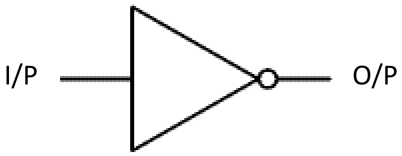
3 - Collector

NOTES:

Date:

NOT GATE IC

Symbol:



Function:

Inversion

I/O Table:

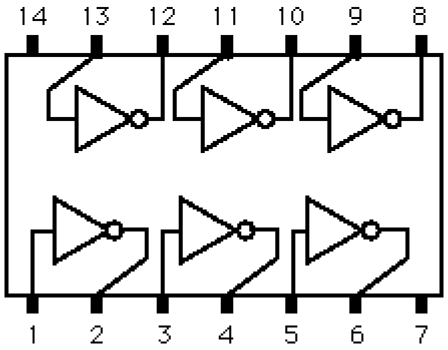
INPUT	OUTPUT
0	1
1	0

IC: 7404 IC (hex-inverter)

Mark the Vin – 14th pin,

Ground – 7th Pin,

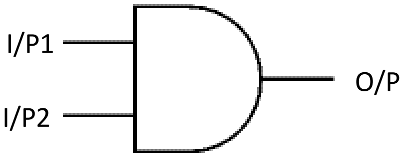
Mark the notch for the IC



Date:

AND GATE IC

Symbol:



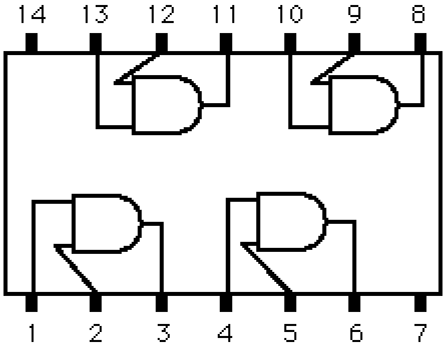
Function:

Digital Multiplication

I/O Table:

INPUT		OUTPUT
0	0	0
0	1	0
1	0	0
1	1	1

IC: 7408 IC (quad 2-Input AND Gate IC)



Mark the Vin – 14th pin,

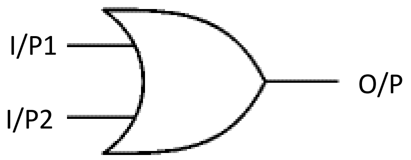
Ground – 7th Pin,

Mark the notch for the IC

Date:

OR GATE IC

Symbol:



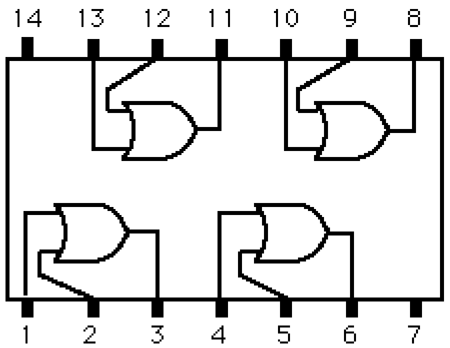
Function:

Digital Addition

I/O Table:

INPUT		OUTPUT
0	0	0
0	1	1
1	0	1
1	1	1

IC: 7432 IC (quad 2-Input OR Gate IC)



Mark the Vin – 14th pin,

Ground – 7th Pin,

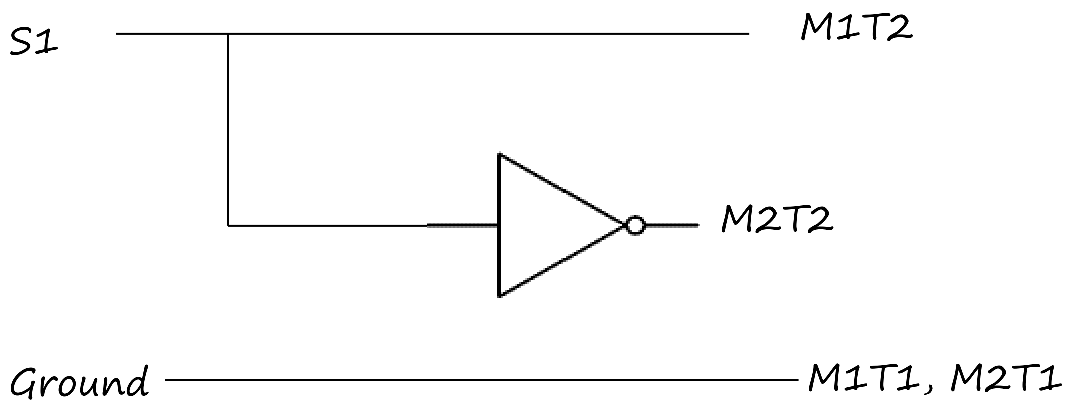
Mark the notch for the IC

Date:

Line Tracer (with 1 sensor):

S1	M1		M2	
	T1	T2	T1	T2
0	0	0	0	1
1	0	1	0	0

Logic-Gates Connection:

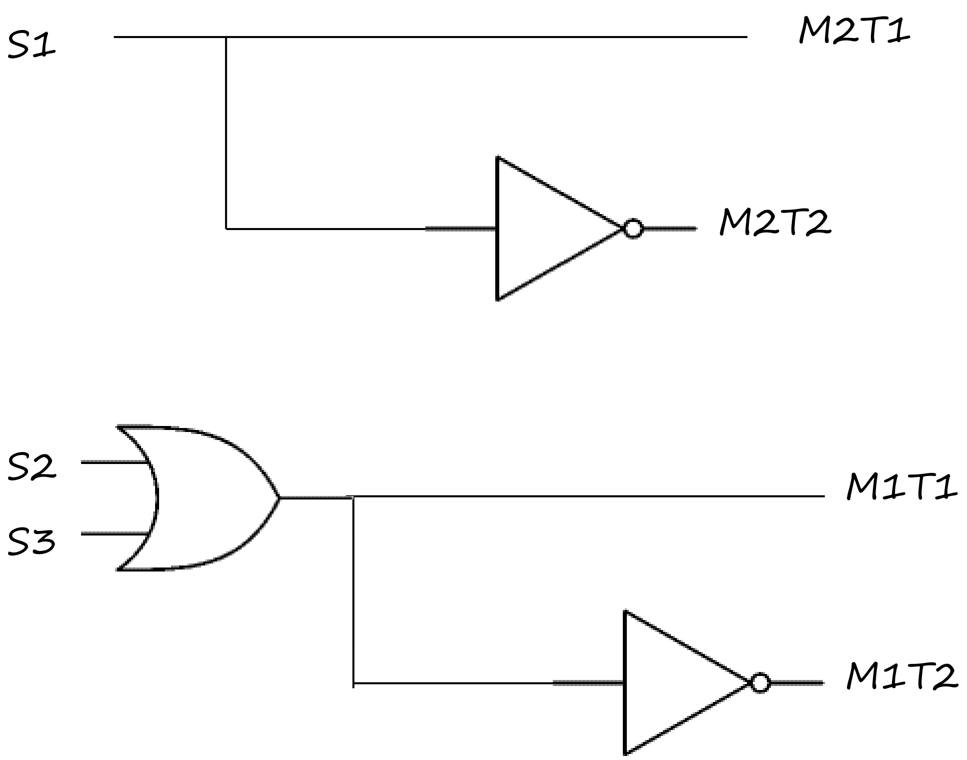


Date:

Obstacle Avoider (with 3 sensors):

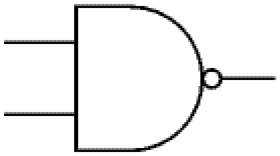
S1	S2	S3	M1		M2	
			T1	T2	T1	T2
0	0	0	0	1	0	1
0	0	1	1	0	0	1
1	0	0	0	1	1	0
0	1	0	1	0	0	1

Logic-Gates Connection:

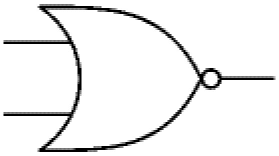


NOTES:

NAND (opposite of AND Gate)

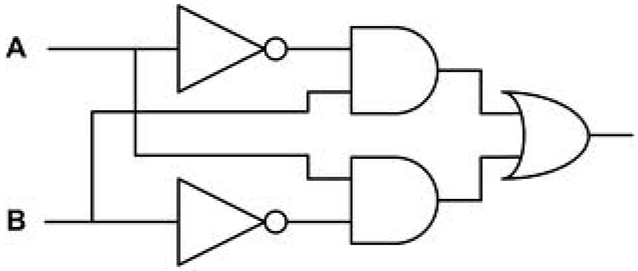


NOR (opposite of OR Gate)

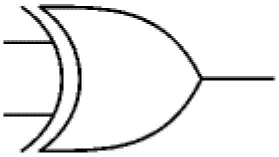


Exercise

Find the input/ouput table



XOR GATE



Gives output 1 when atleast one input is different from the others (or gives output 0 when all the inputs are same)

<u>Note: The above exercise is the Derivation of XOR Gate</u>

Microcontroller:

What is microcontroller?

A **microcontroller** (sometimes abbreviated **µC**, **uC** or **MCU**) is a small computer on a single IC containing a microprocessor, memory, and programmable input/output ports

Features of a Microcontroller

Clock: The clock determines the speed at which the processor works (the speed at which one instruction is executed). It is represented in frequency.

Memory: Mainly two types –

RAM (Random Access Memory) – it affects the speed of the processor. It can be read and erased and written again as many times as required, however it gets emptied at the start of the processor.

ROM (Read only Memory) – does not get erased at the start of the processor.

Flash Memory – Wherein the programs in a microcontroller are stored

I/O Ports: Pins to interface with the external devices

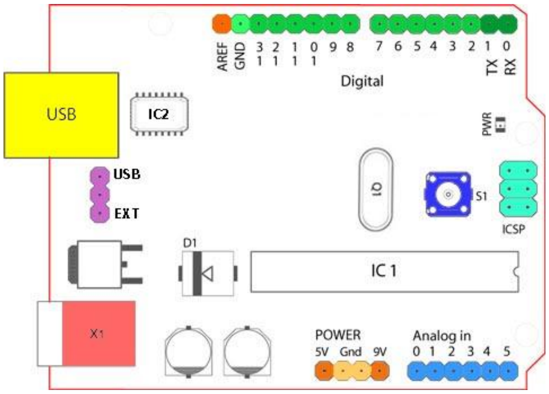
Atmega 328

Atmega is the name of the series of microcontrollers released by ATMEL. “32” represents the Flash Memory size and “8” represents the bits that can be processed at a time

SPDUINO

SPDuino is the microcontroller development board that helps in using the microcontroller directly and interfacing with other I/O devices.
IC1 – ATMEGA 328 (microcontroller)
S1 – reset switch (that resets the program pointer back to the initial position and clears the RAM)
Q1 – Crystal for the clock (16MHz), determines the processor speed)
X1 – Power for the SPDuino
D1 – Diode
USB – for communication with the microcontroller
IC2 – FT232 IC for translation of the USB language into microcontroller understandable language
PWR – Power indicator LED
Jumper next to USB – to select how the microcontroller board needs to be powered – through USB supply or through external supply.

Digital Pins – 14 pins capable of functioning as digital input or output interface.



Analog Pins- 6 pins capable of Analog input (has internal ADC – analog to digital conversion)

Power Pins – pins wherein different voltage levels come as output – can be used if required.

Programming Basics:

The programming is done in 'C' language.

The syntax – the format in which the program has to be typed so that the microcontroller understands it.

The compiler – the error checker which checks for any syntax errors in the program typed

The COM Port – the port through which the microcontroller is connected to your laptop (it will display as FT232)

Terminating a statement - Every instruction/statement should be terminated by a semi-colon, to make the program pointer understand that the instruction is over

Syntax:

Declaring Variables:

Variables of two types – integer (int i;) and character (chac i;)
The integer variables can be assigned with any values
int i = 0;
j=0;
Multiple variables can be declared like this:
int i, j=0,k;

Functions:

Every statement/instruction that needs to be executed should be inside a function.

```
void setup( )
{
….. (will be executed only once at the start of the controller)
}
void loop( )
{
….. (will be executed continuously by the controller)
}
```

Setting a pin as INPUT/OUTPUT:

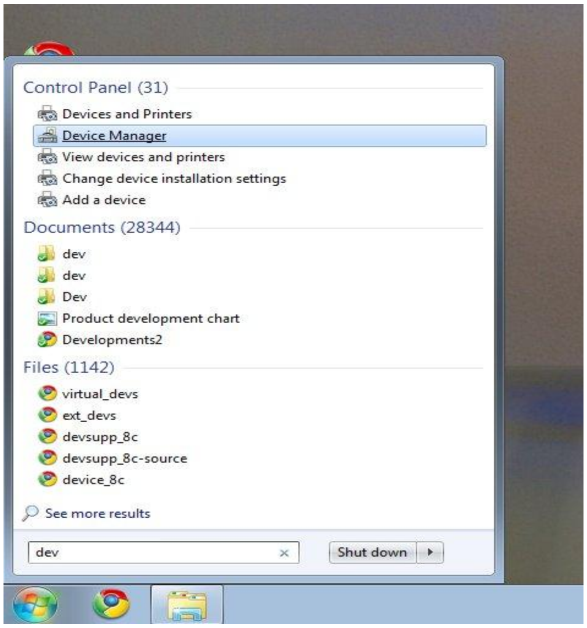
```
pinMode(pin_number, INPUT/OUTPUT);
```

WRITING AN OUTPUT TO A DIGITAL PIN:

```
digitalWrite(pin_number, LOW/HIGH);
```

INSTALLING DRIVER SOFTWARE

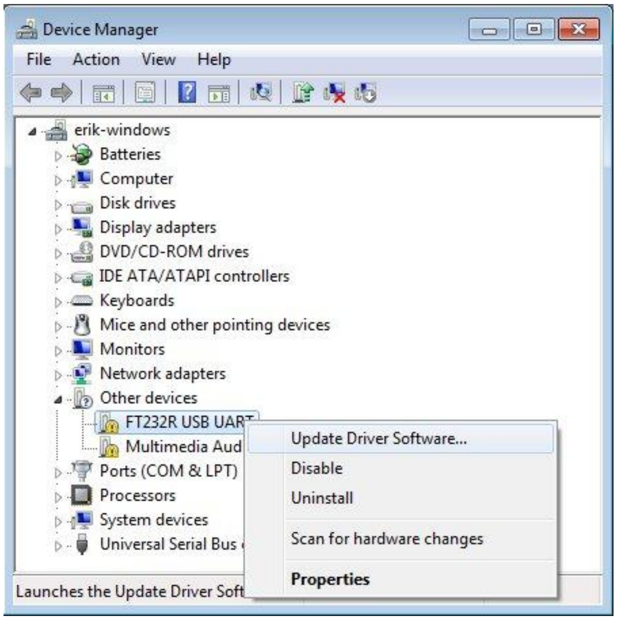
Step 1: Open Device manager



For Windows XP,

- Go to my-computer
- Right Click
- Choose SYSTEM PROPERTIES
- Choose HARDWARE Tab
- Click DEVICE MANAGER

Step 2: FT232 USB

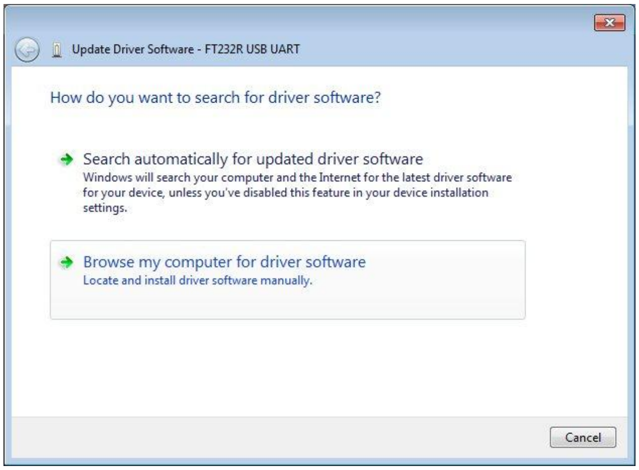


Find an exclamatory marked device named FT232.

Right click and click update driver software

Step 3: Update Driver

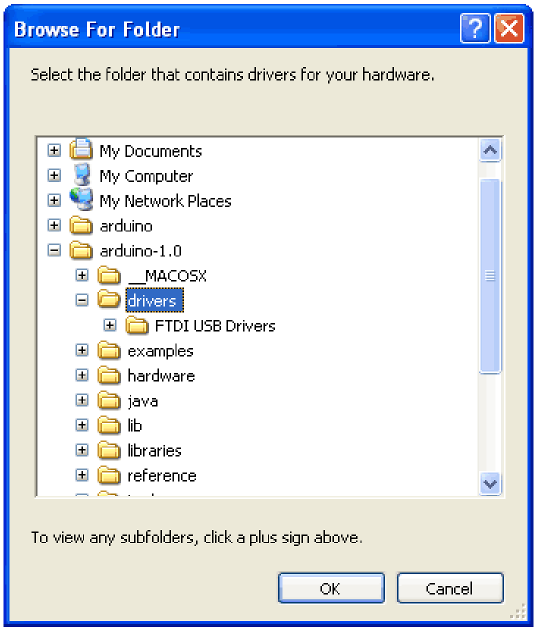
- Select Browse my computer
- Choose the drivers folder under the extracted arduino file
- Click ok
- Then click next
- If you see an exclamatory marked device named USB Serial Port, continue the process till you get a COM Port number


Update Driver Software - FT232R USB UART

How do you want to search for driver software?

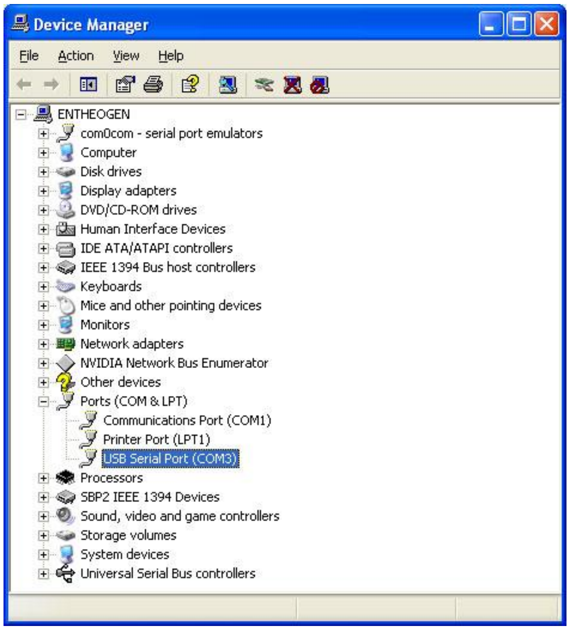
Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

Browse my computer for driver software
Locate and install driver software manually.


Browse For Folder

Select the folder that contains drivers for your hardware.

To view any subfolders, click a plus sign above.



SELECTING BOARD AND COM PORT

In the Arduino IDE, Select the following before uploading,

- TOOLS > BOARDS > ARDUINO DEUMILANOVE W/ATMEGA328
- TOOLS > SERIAL PORT > PORT NUMBER

EXP1_LED_GLOWING

```
void setup( )
{
pinMode(13,OUTPUT);
}
void loop( )
{
digitalWrite(13,HIGH);
}
```

EXP2_LED_BLINKING

```
void setup( )
{
pinMode(13,OUTPUT);
}
void loop( )
{
digitalWrite(13,HIGH);
delay(1000);
digitalWrite(13,LOW);
delay(1000);
}
```

When a delay function is included in a program, the program pointer freezes at that point for the delay period mentioned. Reading of inputs does not occur, outputs in all the pins remain the same for the delay period.

EXP3_LED_SEQUENCING

5 LEDs are connected in the following order:

1st LED – 2(+) and 3(-)
2nd LED – 4(+) and 5(-)
3rd LED – 6(+) and 7(-)
4th LED – 8(+) and 9(-)
5th LED – 10(+) and 11(-)

```
void setup( )
{
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);
pinMode(7,OUTPUT);
pinMode(8,OUTPUT);
pinMode(9,OUTPUT);
pinMode(10,OUTPUT);
pinMode(11,OUTPUT);

//since these pins are always LOW
digitalWrite(3,LOW);
digitalWrite(5,LOW);
digitalWrite(7,LOW);
digitalWrite(9,LOW);
digitalWrite(11,LOW);

}
```

```
void loop( )
{
digitalWrite(2,HIGH);
delay(100);
digitalWrite(2,LOW);
digitalWrite(4,HIGH);
delay(100);
digitalWrite(4,LOW);
digitalWrite(6,HIGH);
delay(100);
digitalWrite(6,LOW);
digitalWrite(8,HIGH);
delay(100);
digitalWrite(8,LOW);
digitalWrite(10,HIGH);
delay(100);
digitalWrite(10,LOW);
}
```

Date:

EXP4_LED_SEQUENCING_REVERSE

```
void setup( )
{
pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);
pinMode(7,OUTPUT);
pinMode(8,OUTPUT);
pinMode(9,OUTPUT);
pinMode(10,OUTPUT);
pinMode(11,OUTPUT);

//since these pins are always LOW
digitalWrite(3,LOW);
digitalWrite(5,LOW);
digitalWrite(7,LOW);
digitalWrite(9,LOW);
digitalWrite(11,LOW);

}
```

```
void loop( )
{
digitalWrite(10,HIGH);
delay(100);
digitalWrite(10,LOW);
digitalWrite(8,HIGH);
delay(100);
digitalWrite(8,LOW);
digitalWrite(6,HIGH);
delay(100);
digitalWrite(6,LOW);
digitalWrite(4,HIGH);
delay(100);
digitalWrite(4,LOW);
digitalWrite(2,HIGH);
delay(100);
digitalWrite(2,LOW);
}
```

Date:

EXP5_SS_INPUT_SWITCH_LED

```
int value;
void setup( )
{
pinMode(A4,INPUT);
pinMode(3,OUTPUT);
}

void loop( )
{
value=digitalRead(A4);
if(value==HIGH)
{
    digitalWrite(3,HIGH);
}
else
{
   digitalWrite(6,LOW);
}
}
```

EXP6_SS_SWITCH_LED_SEQUENCING

```
int value;
void setup( )
{
pinMode(A4,INPUT);

pinMode(2,OUTPUT);
pinMode(3,OUTPUT);
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);
pinMode(7,OUTPUT);
pinMode(8,OUTPUT);
pinMode(9,OUTPUT);
pinMode(10,OUTPUT);
pinMode(11,OUTPUT);
//since these pins are always LOW
digitalWrite(3,LOW);
digitalWrite(5,LOW);
digitalWrite(7,LOW);
digitalWrite(9,LOW);
digitalWrite(11,LOW);
}

void loop( )
{
value=digitalRead(A4);
if(value==HIGH)
{
    EXP_4;
}
else
{
   EXP_5;
}
}
```

EXP7_SS_LED_BRIGHTNESS

```
void setup( )
{
pinMode(3,OUTPUT);
}
void loop( )
{
analogWrite(3,0);
delay(1000);
analogWrite(3,128);
delay(1000);
analogWrite(3,255);
delay(1000);
}
```

EXP8_SS_FOR_LOOP

```
void setup( )
{
pinMode(3,OUTPUT);
}
void loop( )
{
for(i=0; i<255; i=i-1)
{
analogWrite(3,i);
delay(10);
}
for(i=255; i>0; i=i-1)
{
analogWrite(3,i);
delay(10);
}
}
```

EXP9_SS_SENSOR_RELAY

```
int value;
void setup( )
{
pinMode(A0,INPUT);
pinMode(A5,OUTPUT);
}

void loop( )
{
value=digitalRead(A0);
if(value==HIGH)
{
    digitalWrite(A5,HIGH);
}
else
{
   digitalWrite(A5,LOW);
}
}
```

Date:

EXP10_SERIAL_COMMUNICATION

```
void setup( )
{
Serial.begin(9600);
pinMode(A0,INPUT);
}

void loop()
{
value=digitalRead(A0);
Serial.println("sensor value is");
Serial.println(value);
}
```

Date:

EXP11_SWITCH_CASE

```
void setup( )
{
Serial.begin(9600);
Serial.println("Welcome to KIDOBOTIKZ");
Serial.println("Press '1' for the menu ");
}

void loop()
{
value=Serial.read( );
switch(value)
{
case '1':
Serial.println("Press 2 for attendance");
Serial.println("Press 3 for results");
Serial.println("Press 4 for competition details");
break;
case '2':
Serial.println("Full attendance, updated till 21-10-2013 ");
Serial.println("Press 1 for main menu");
Serial.println("Press 0 for exit");
break;
case '3':
Serial.println("Results: 47/50 in Beginner Level Theory exam ");
Serial.println("Press 1 for main menu");
Serial.println("Press 0 for exit");
break;
case '4':
Serial.println("No upcoming competitions");
Serial.println("Press 1 for main menu");
Serial.println("Press 0 for exit");
break;
case '0':
Serial.println("Thank you. Have a good day");
break;
}
}
```

Date:

EXP12_SS_LCD_DISPLAY

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,8,7,4,2);
void setup( )
{
  lcd.begin(16, 2);
}
void loop( )
{
  lcd.setCursor(0, 0);
  lcd.print("kidobotikz");
  lcd.setCursor(0, 1);
  lcd.print("sp robotic works");
}
```

Date:

EXP13_SS_ANALOG_INPUT

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,8,7,4,2);
int value;
void setup( )
{
  lcd.begin(16, 2);
  pinMode(A3,INPUT);
}
void loop( )
{
  value = analogRead(A3);
  lcd.clear( );
  lcd.setCursor(0, 0);
  lcd.print("LDR Value");
  lcd.setCursor(0, 1);
  lcd.print(value);
}
```

Date:

EXP14_SS_MOTOR_DIR_SPD_CONTROL

```
void setup( )
{
pinMode(5,OUTPUT);
pinMode(6,OUTPUT);
}


void loop( )
{
analogWrite(5,255);
analogWrite(6,0);
delay(1000);
analogWrite(5,128);
analogWrite(6,0);
delay(1000);
analogWrite(5,0);
analogWrite(6,0);
delay(1000);

analogWrite(5,0);
analogWrite(6,255);
delay(1000);
analogWrite(5,0);
analogWrite(6,128);
delay(1000);
analogWrite(5,0);
analogWrite(6,0);
delay(1000);
}
```

EXP15_SS_IR_REMOTE

```
#include <SPIRremote.h>
SPIRrecv remote(9);
int value;
void setup()
{
  remote.enableIR();
  pinMode(3, OUTPUT);
}
void loop()
{
  value = remote.getIRValue();
  if(value == 1)
  {
    digitalWrite(3,HIGH);
  }
  if (irValue == 0)
  {
    digitalWrite(3,LOW);
  }
}
```