ELECTRONICS CLUB IITKANPUR

UART

RUDRA PRATAP SUMAN

UART: Universal Asynchronous Receiver Transmitter

- UART is a simple half-duplex, asynchronous, serial protocol.
- Simple communication between two equivalent nodes.
- Any node can initiate communication.
- Since connection is half-duplex, the two lanes of communication are completely independent.

ELECTRONICS CLUB IITKANPUR

UART: Universal Asynchronous Receiver Transmitter

- What makes it ‘universal’ ?
  - Its parameters (format,speed ..) are configurable.
- Why ‘asynchronous’ ?
  - It doesn’t have a clock

UART Basics

- Baud Rate:
  - No. of bits transmitted/received per second = ______bits/sec.
- Format of Communication

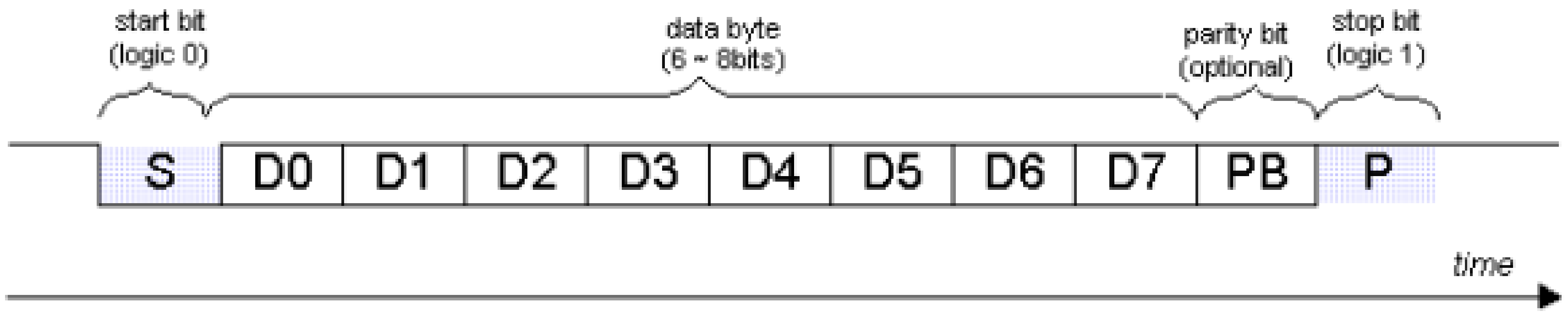


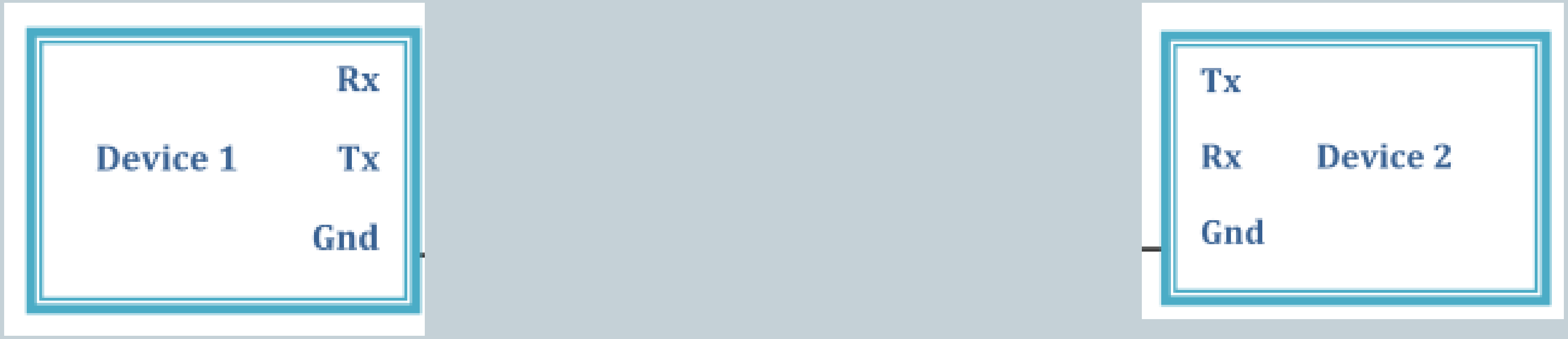
Figure 17: Basic UART packet format: 1 start bit, 8 data bits, 1 parity bit, 1 stop bit.

UART Basics

Connections for UART

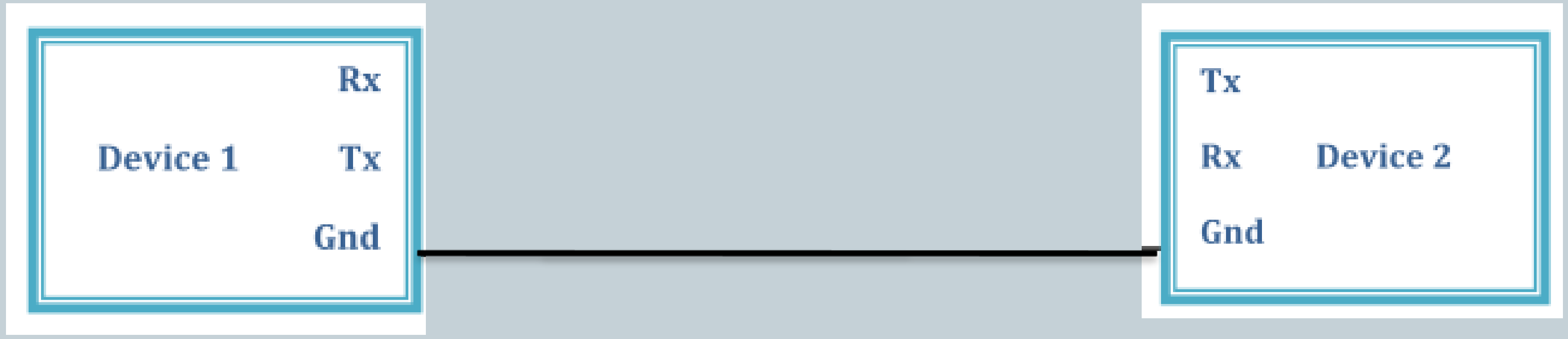
UART Basics

Connections for UART



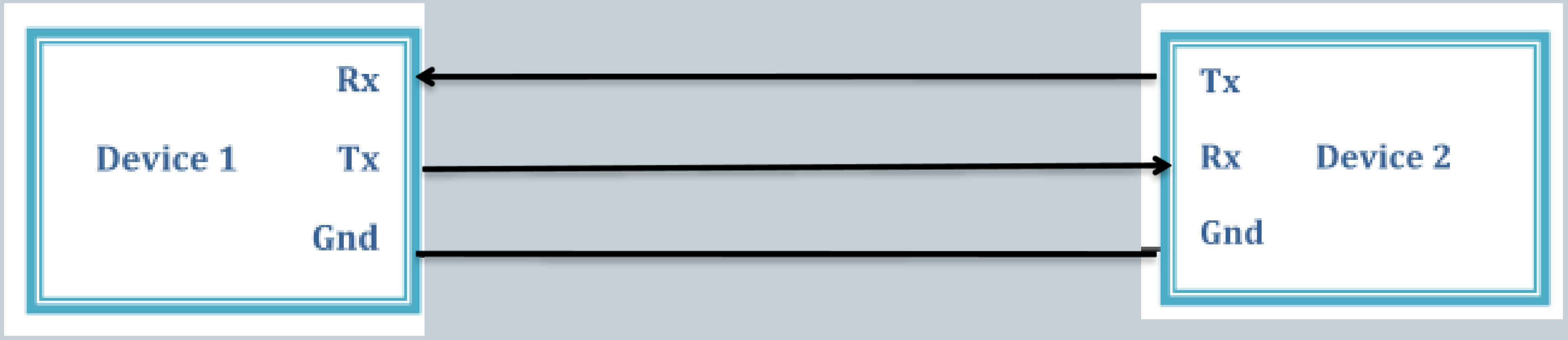
UART Basics

Connections for UART



UART Basics

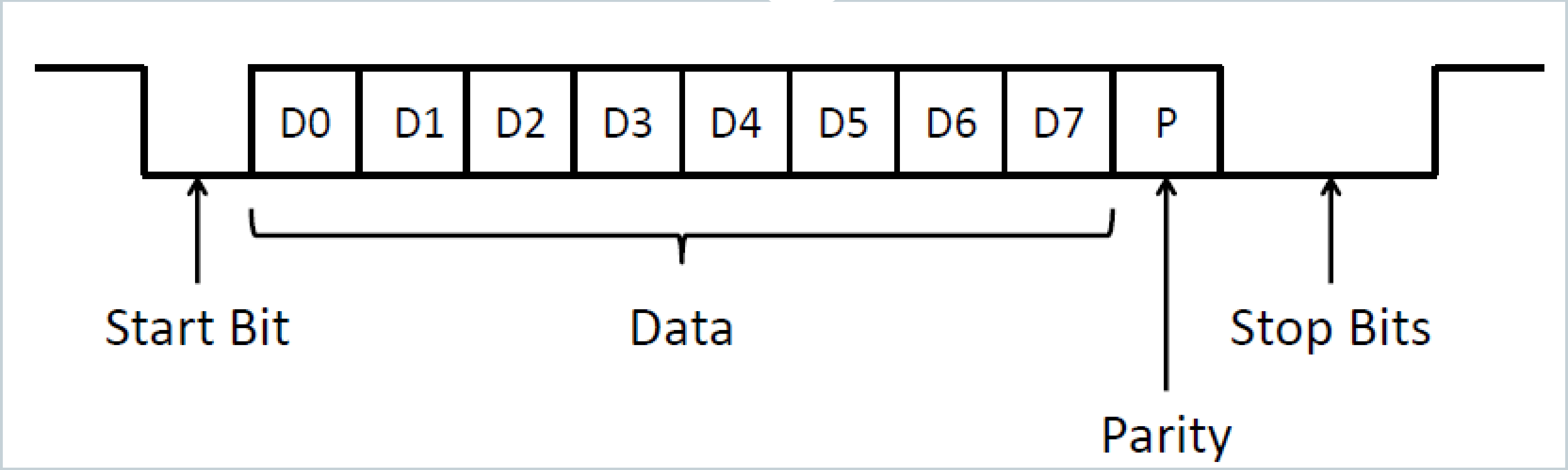
Connections for UART



UART Characteristics

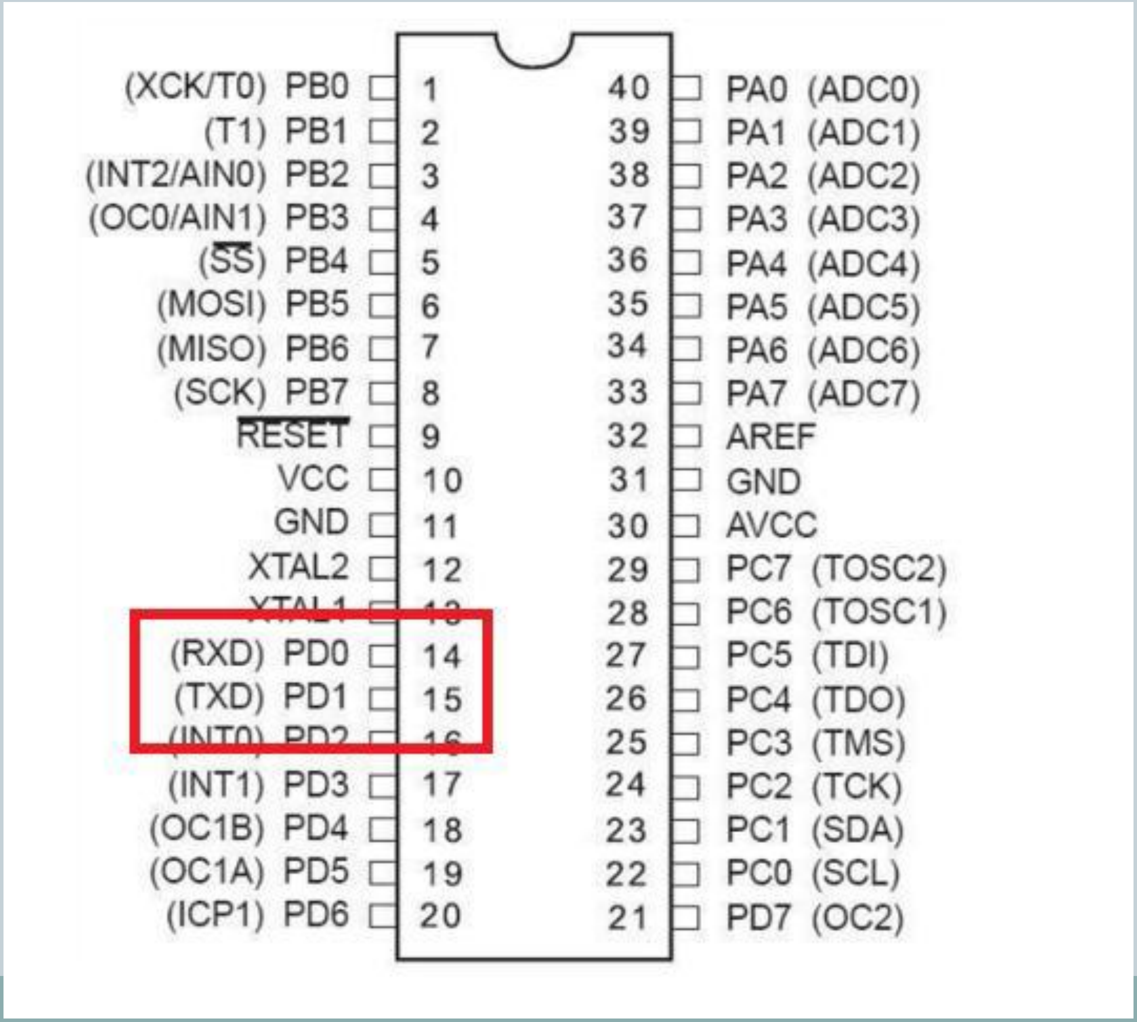
- The speed of communication (measured in bauds) is predetermined on both ends.
- A general rule of thumb is to use 9600 bauds for wired communication.
- UART implements error-detection in the form of parity bit.

Parity Bit

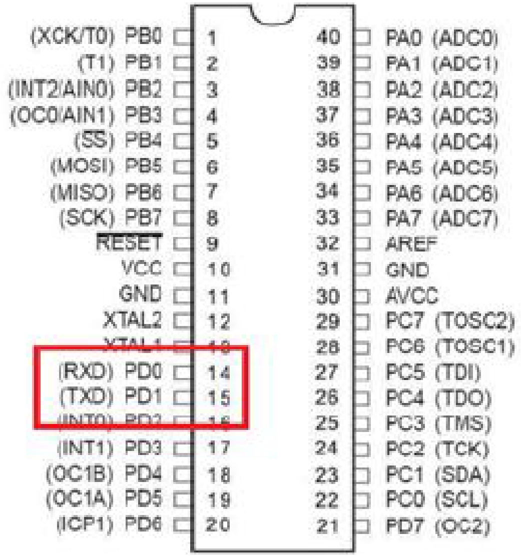


- Parity bit is HIGH when number of 1’s in the Data is odd.
- Respectively, it is LOW when number of 1’s in the Data is even

UART in AtMega16

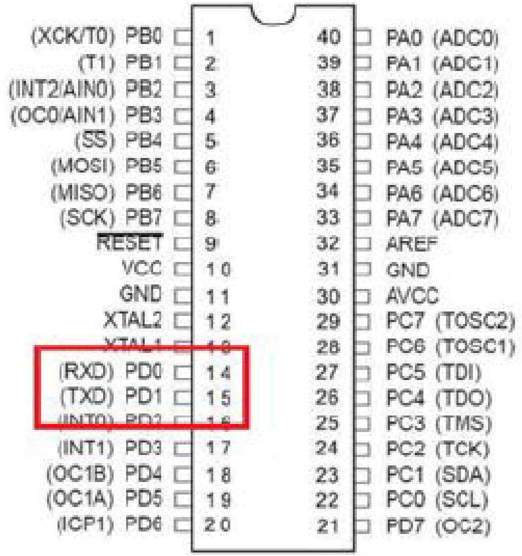


Connecting AtMega16's with UART



Device 1

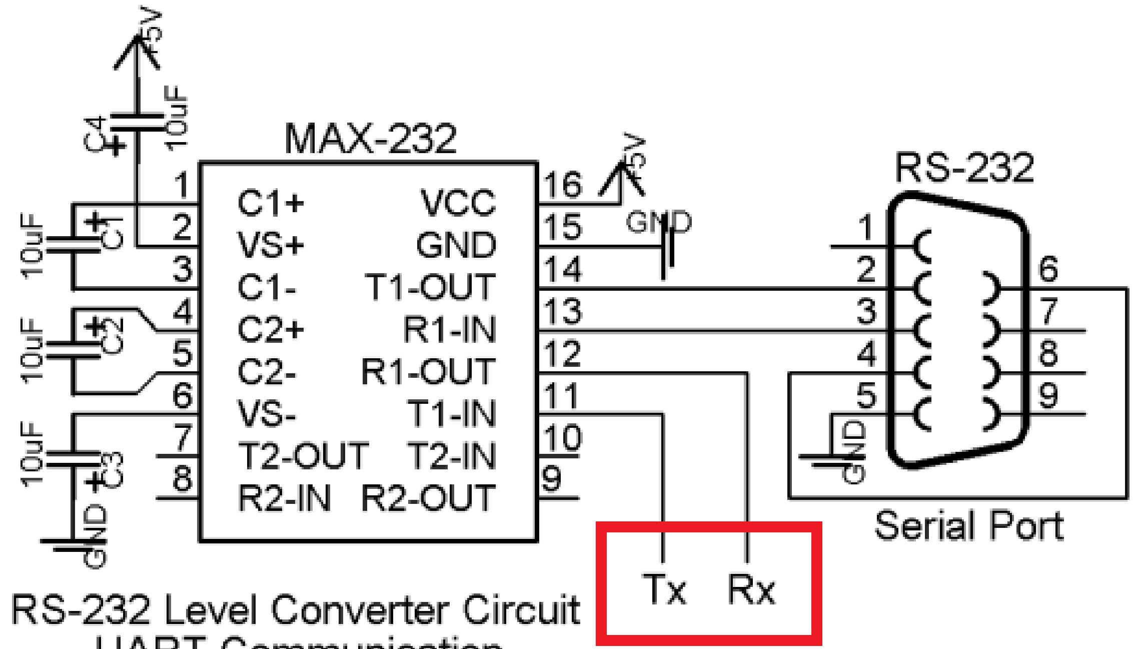
Rx	?	Tx
Tx	?	Rx
GND	?	GND



Device 2

ELECTRONICS CLUB IITKANPUR

MAX-232 and USB-Serial



RS-232 Level Converter Circuit
UART Communication

Connecting AtMega16 with Computer

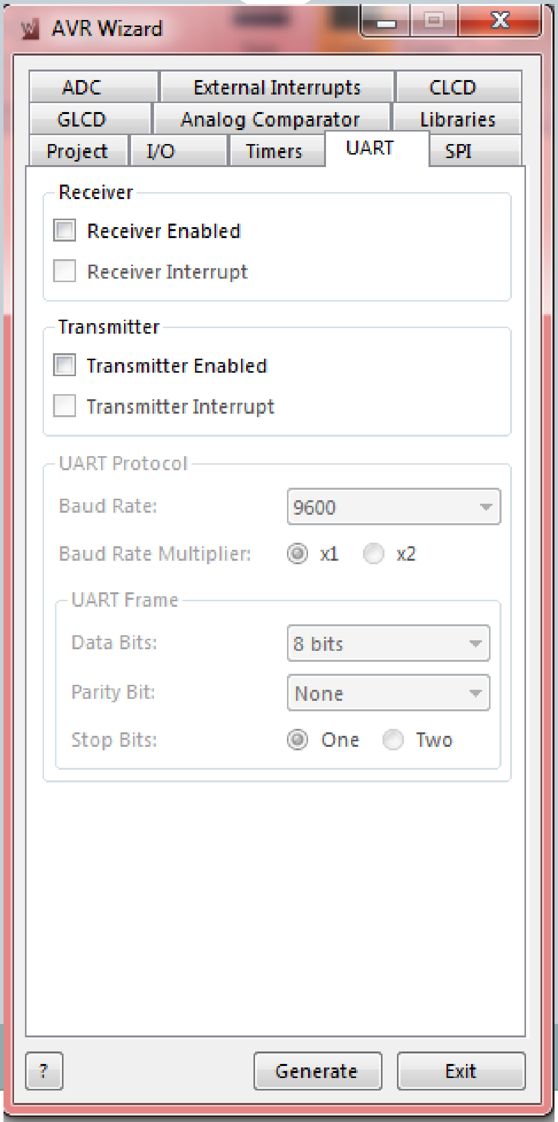
Latest Direct Way :

ELECTRONICS CLUB IITKANPUR

Coding with UART

- Three simple commands :
  - –putchar(char);
  - sends 8-bit characters through UART
  - –getchar();
  - receives 8-bit characters via UART
  - –puts(string);
  - sends a constant string

Where do we code.. ?


AVR Wizard

ADC | External Interrupts | CLCD | GLCD | Analog Comparator | Libraries | Project | I/O | Timers | UART | SPI

Receiver
- [ ] Receiver Enabled
- [ ] Receiver Interrupt

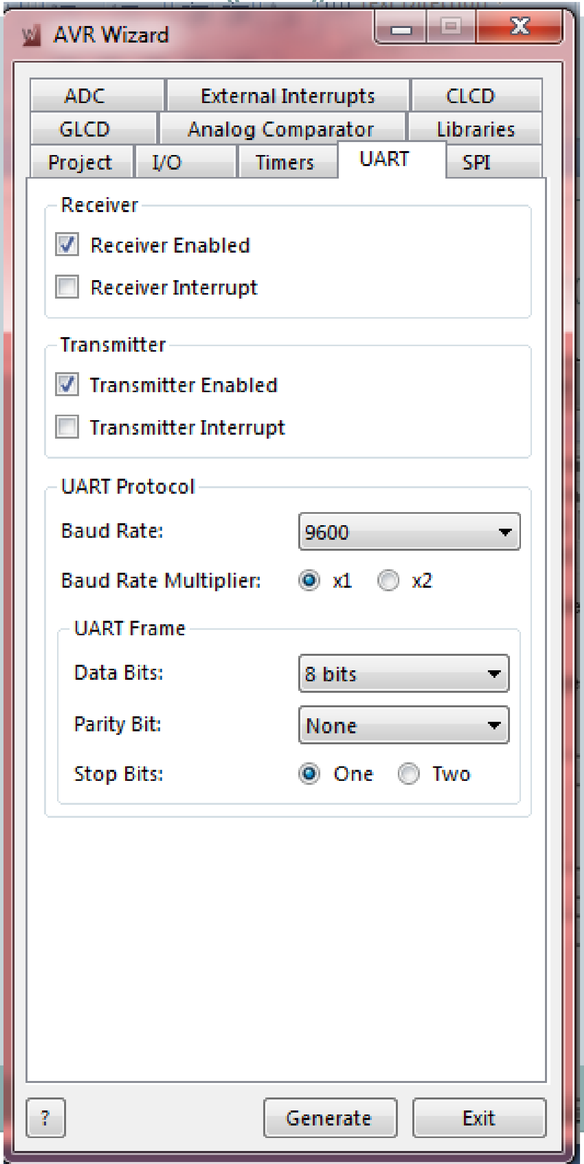
Transmitter
- [ ] Transmitter Enabled
- [ ] Transmitter Interrupt

UART Protocol
- Baud Rate: 9600
- Baud Rate Multiplier: [✓ x1] x2

UART Frame
- Data Bits: 8 bits
- Parity Bit: None
- Stop Bits: [✓ One] Two

ELECTRONICS CLUB IITKANPUR

Where do we code.. ?



Receiver
- [✓] Receiver Enabled
- [ ] Receiver Interrupt

Transmitter
- [✓] Transmitter Enabled
- [ ] Transmitter Interrupt

UART Protocol

Baud Rate: 9600

Baud Rate Multiplier: [✓ x1] x2

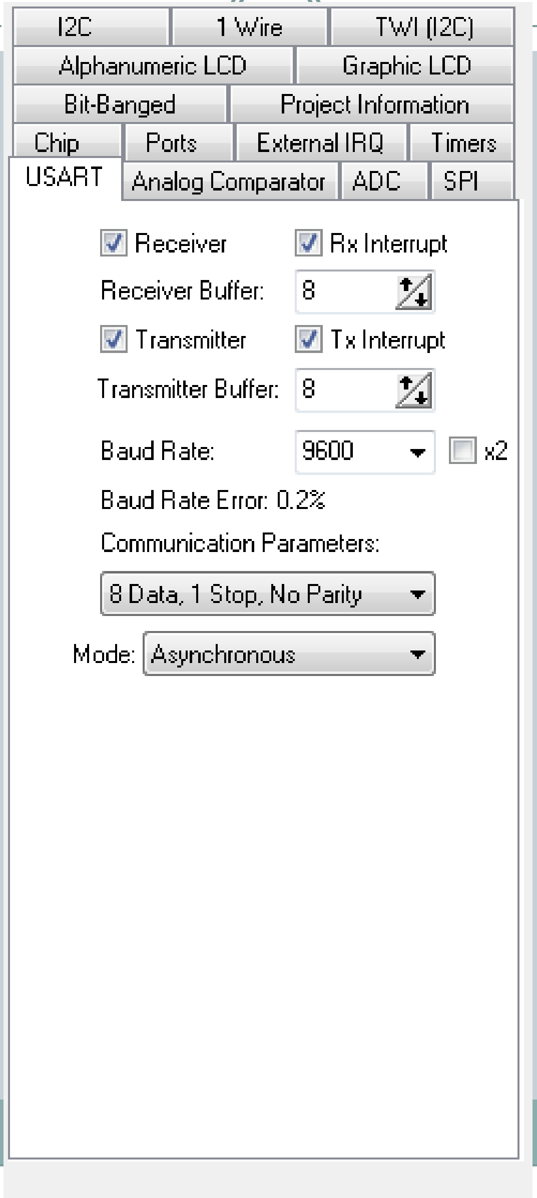
UART Frame

Data Bits: 8 bits

Parity Bit: None

Stop Bits: [✓ One] Two

Where do we code.. ?



☑ Receiver ☑ Rx Interrupt

Receiver Buffer: 8

☑ Transmitter ☑ Tx Interrupt

Transmitter Buffer: 8

Baud Rate: 9600 ☐ x2

Baud Rate Error: 0.2%

Communication Parameters:

8 Data, 1 Stop, No Parity

Mode: Asynchronous

Sample Code for UART

Input MCU

```
 // a is a char variable
a = inputFromUser();
putchar(a);  // Data transmitted, now print
```

LCD MCU

```
a = getchar();
// Program will wait for data

 printChar(a);
```

Coding for Arduino

Coding for Arduino

- Serial.begin(speed)
  - Sets the data rate in bits per second (baud) for serial data transmission.

ELECTRONICS CLUB IITKANPUR

Coding for Arduino

- Serial.begin(speed)
  - Sets the data rate in bits per second (baud) for serial data transmission.
- Serial.end()
  - Disables serial communication, allowing the RX and TX pins to be used for general input and output.
  - To re-enable serial communication, call Serial.begin().

Coding for Arduino

- Serial.begin(speed)
  - Sets the data rate in bits per second (baud) for serial data transmission.
- Serial.end()
  - Disables serial communication, allowing the RX and TX pins to be used for general input and output.
  - To re-enable serial communication, call Serial.begin().
- Serial.read()
  - Reads incoming serial data

Coding for Arduino

- Serial.begin(speed)
  - Sets the data rate in bits per second (baud) for serial data transmission.
- Serial.end()
  - Disables serial communication, allowing the RX and TX pins to be used for general input and output.
  - To re-enable serial communication, call Serial.begin().
- Serial.read()
  - Reads incoming serial data
- Serial.println(val)
  Serial.println(val, format)
  - Prints data to the serial port as human-readable ASCII text followed by a carriage return character (ASCII 13, or '\r') and a newline character (ASCII 10, or '\n')

Coding for Arduino

- Serial.print(val)
  Serial.print(val, format)
  - Prints data to the serial port as human-readable ASCII text.

ELECTRONICS CLUB IITKANPUR

Coding for Arduino

- Serial.print(val)
  Serial.print(val, format)
  - Prints data to the serial port as human-readable ASCII text.
- Serial.flush()
  - Waits for the transmission of outgoing serial data to complete. (Prior to Arduino 1.0, this instead removed any buffered incoming serial data.)

Coding for Arduino

- Serial.print(val)
  Serial.print(val, format)
  - Prints data to the serial port as human-readable ASCII text.
- Serial.flush()
  - Waits for the transmission of outgoing serial data to complete. (Prior to Arduino 1.0, this instead removed any buffered incoming serial data.)
- Serial.available()
  - Get the number of bytes (characters) available for reading from the serial port. This is data that's already arrived and stored in the serial receive buffer (which holds 64 bytes).

Sample Code for Arduino

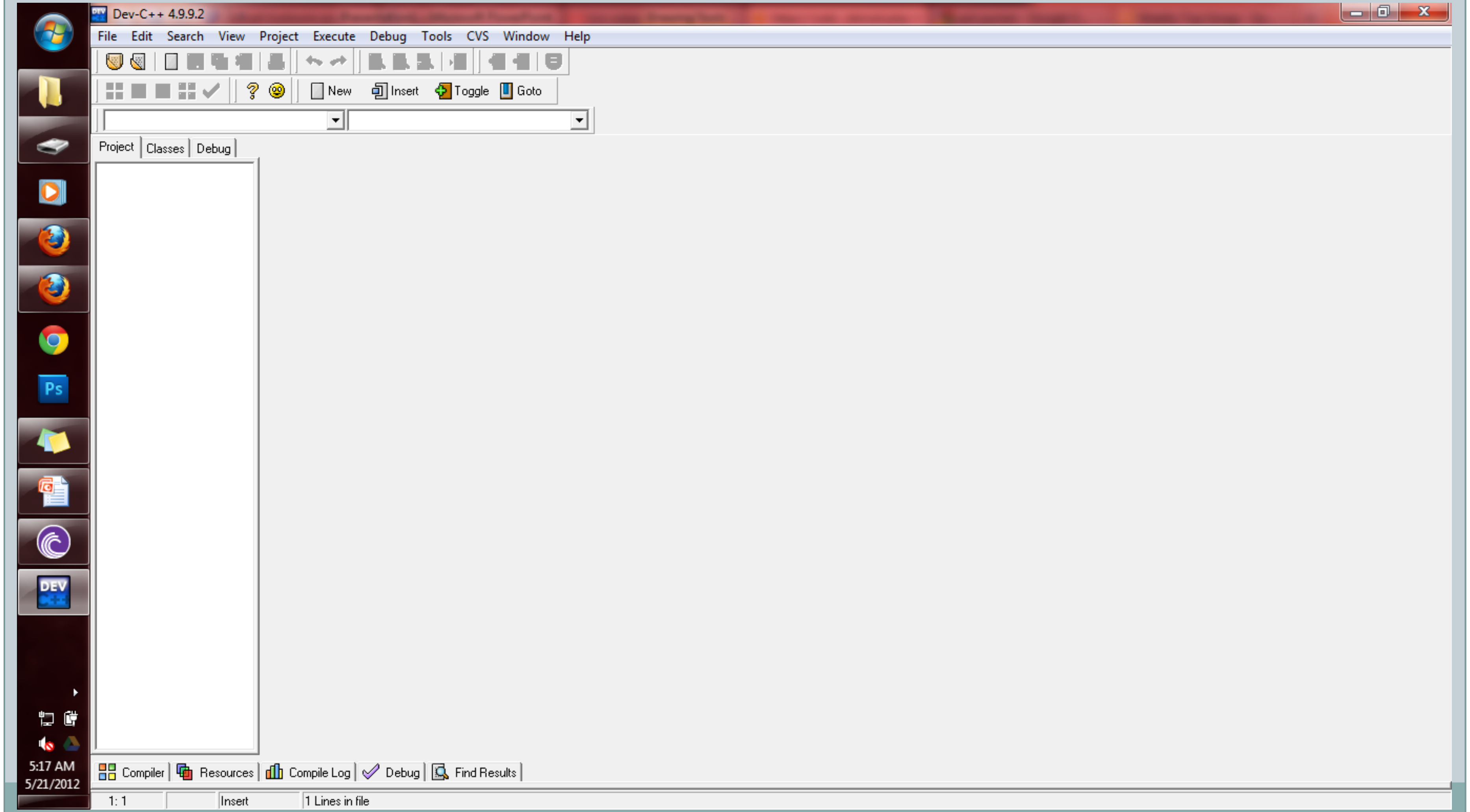
```
int incomingByte = 0;   // for incoming serial data

void setup() {
        Serial.begin(9600);     // opens serial port, sets data rate to
9600 bps
}

void loop() {

        // send data only when you receive data:
        if (Serial.available() > 0) {
                // read the incoming byte:
                incomingByte = Serial.read();

                // say what you got:
                Serial.print("I received: ");
                Serial.println(incomingByte, DEC);
        }
}
```

Coding in DevCPP

link for downloading DevC++

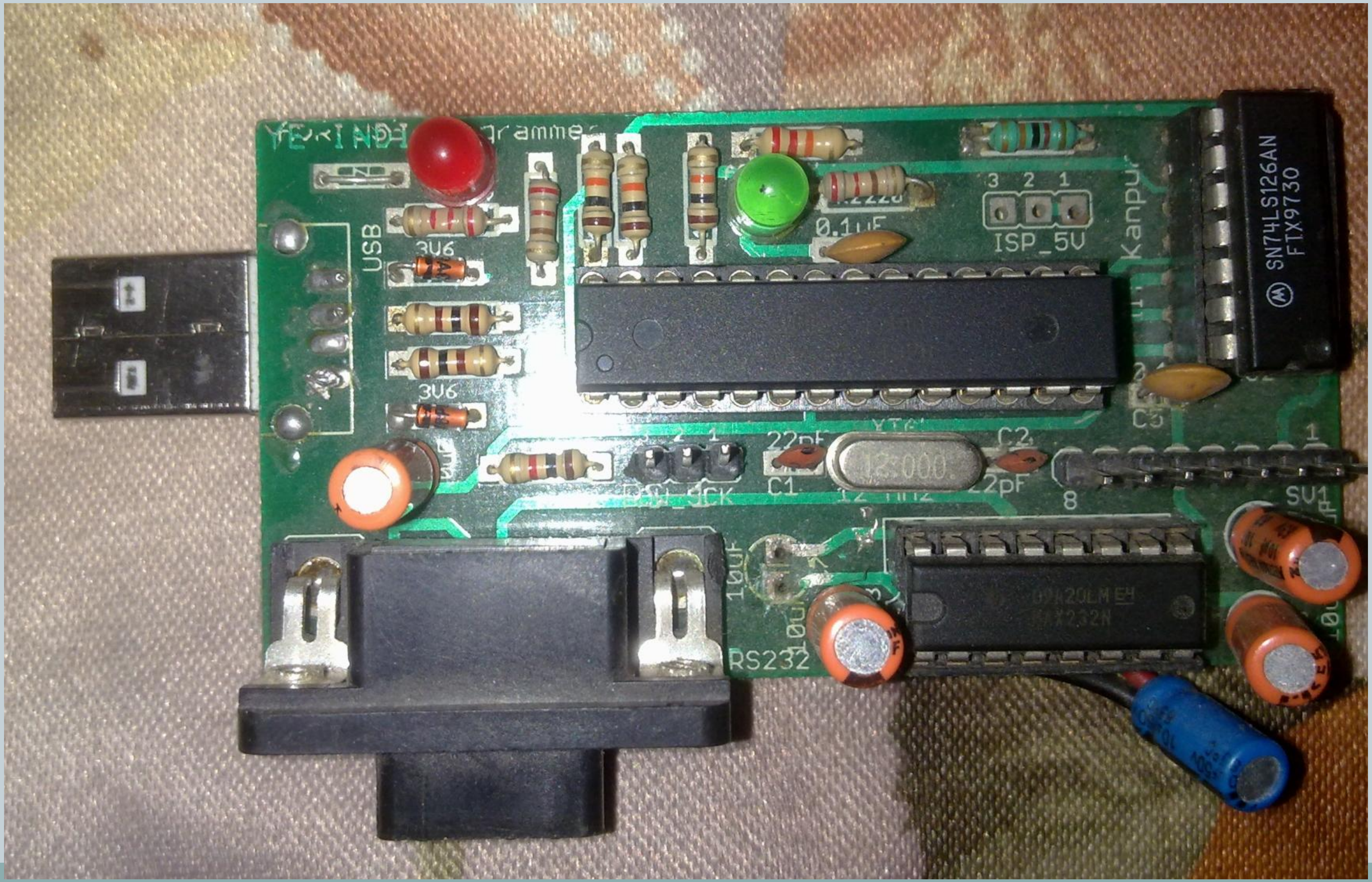
http://sourceforge.net/projects/dev-cpp/files/Binaries/Dev-C%2B%2B%204.9.9.2/devcpp-4.9.9.2_setup.exe/download

Coding in DevCPP

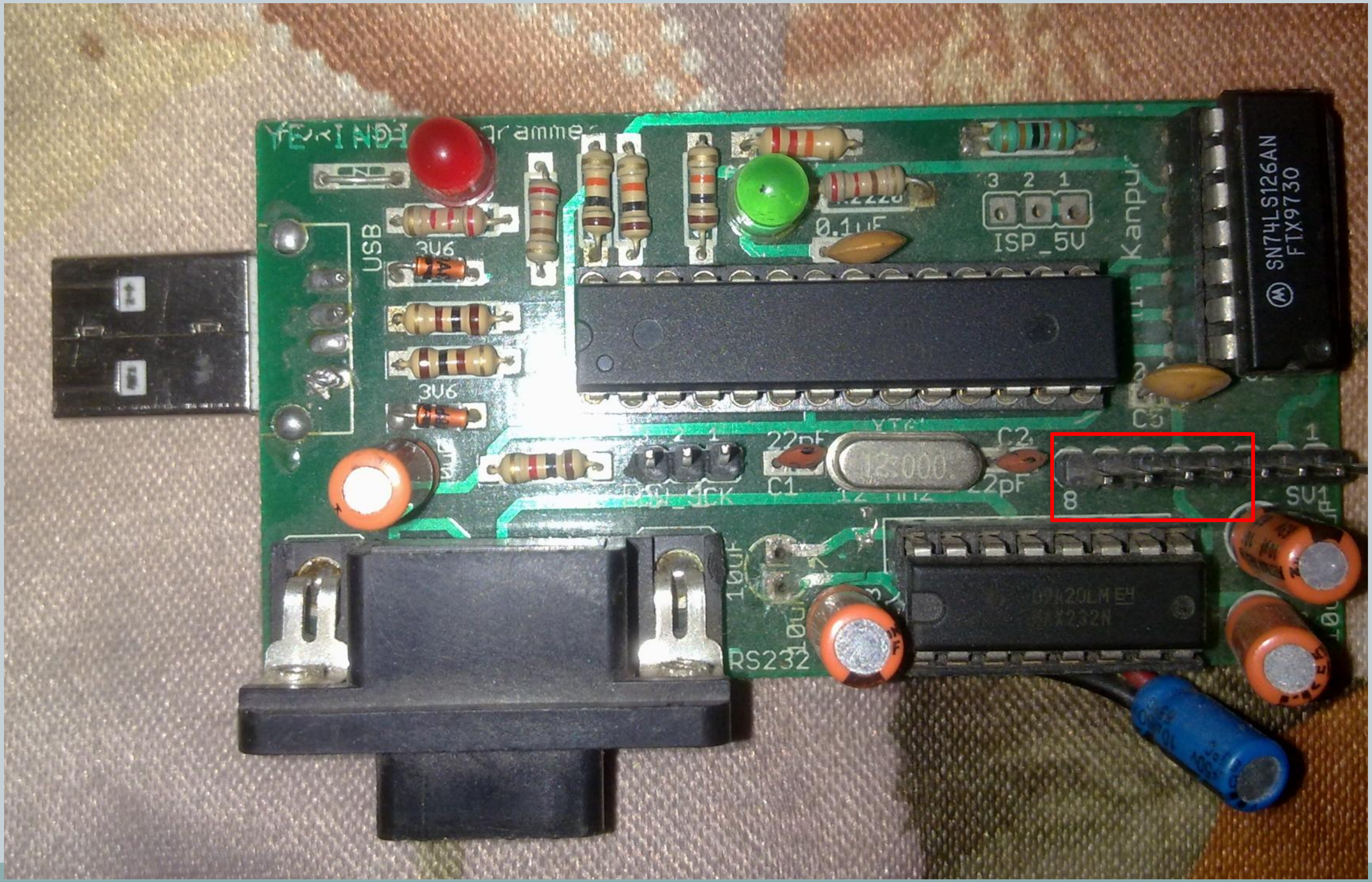


Dev-C++ 4.9.9.2
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
Project Classes Debug
Compiler Resources Compile Log Debug Find Results
1: 1 Insert 1 Lines in file
5:17 AM
5/21/2012

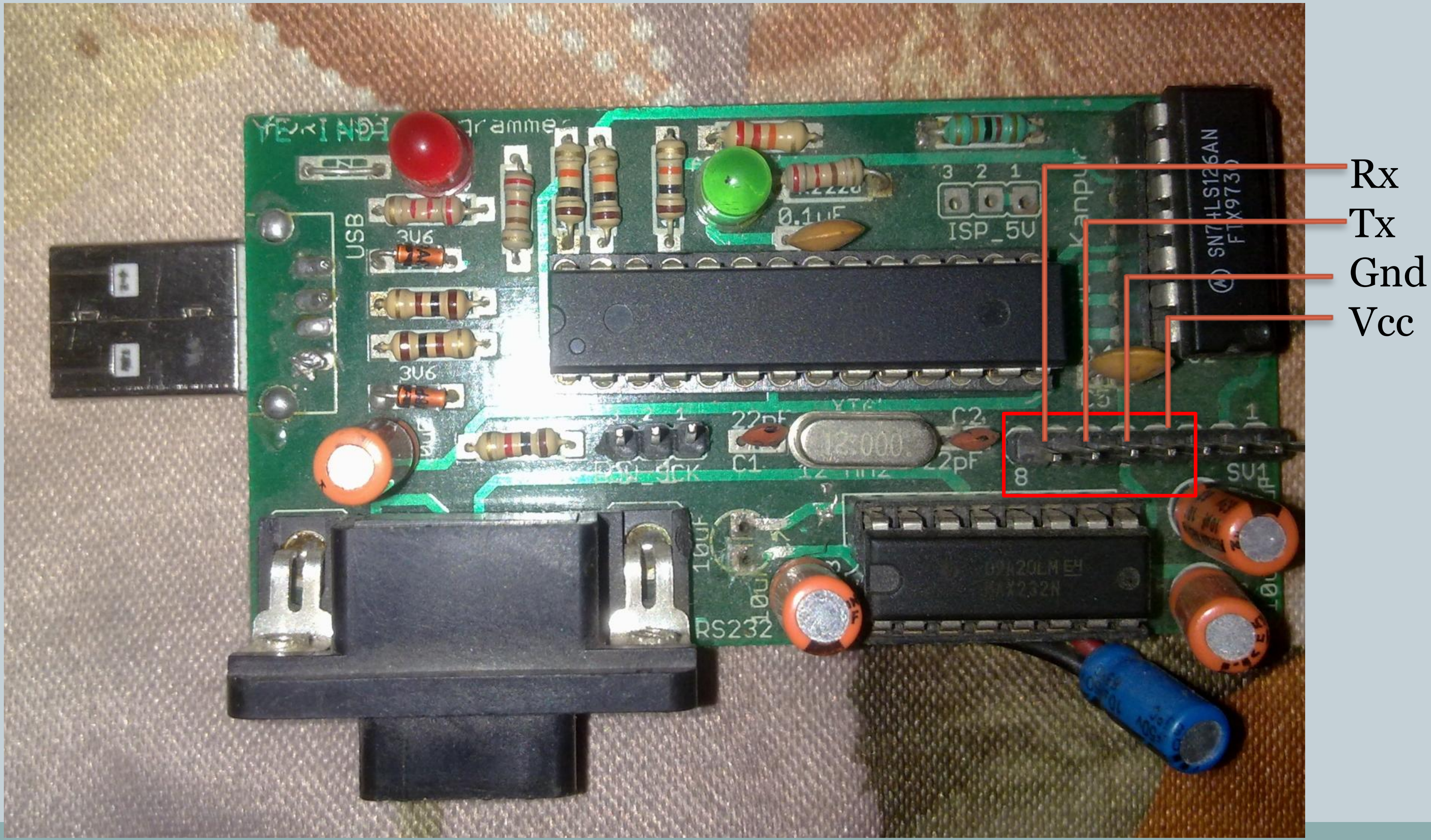
Coding in DevCPP



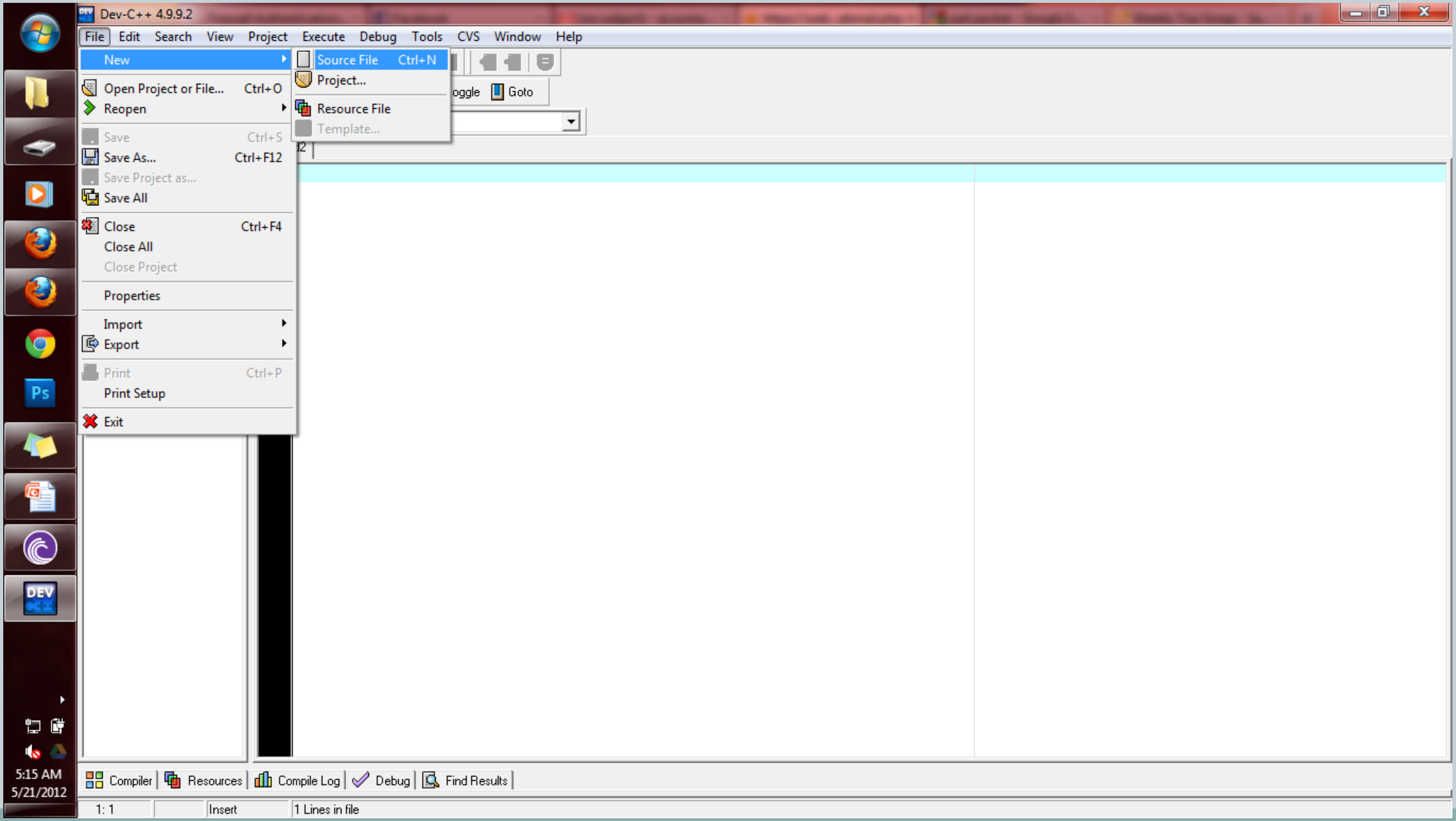
Coding in DevCPP



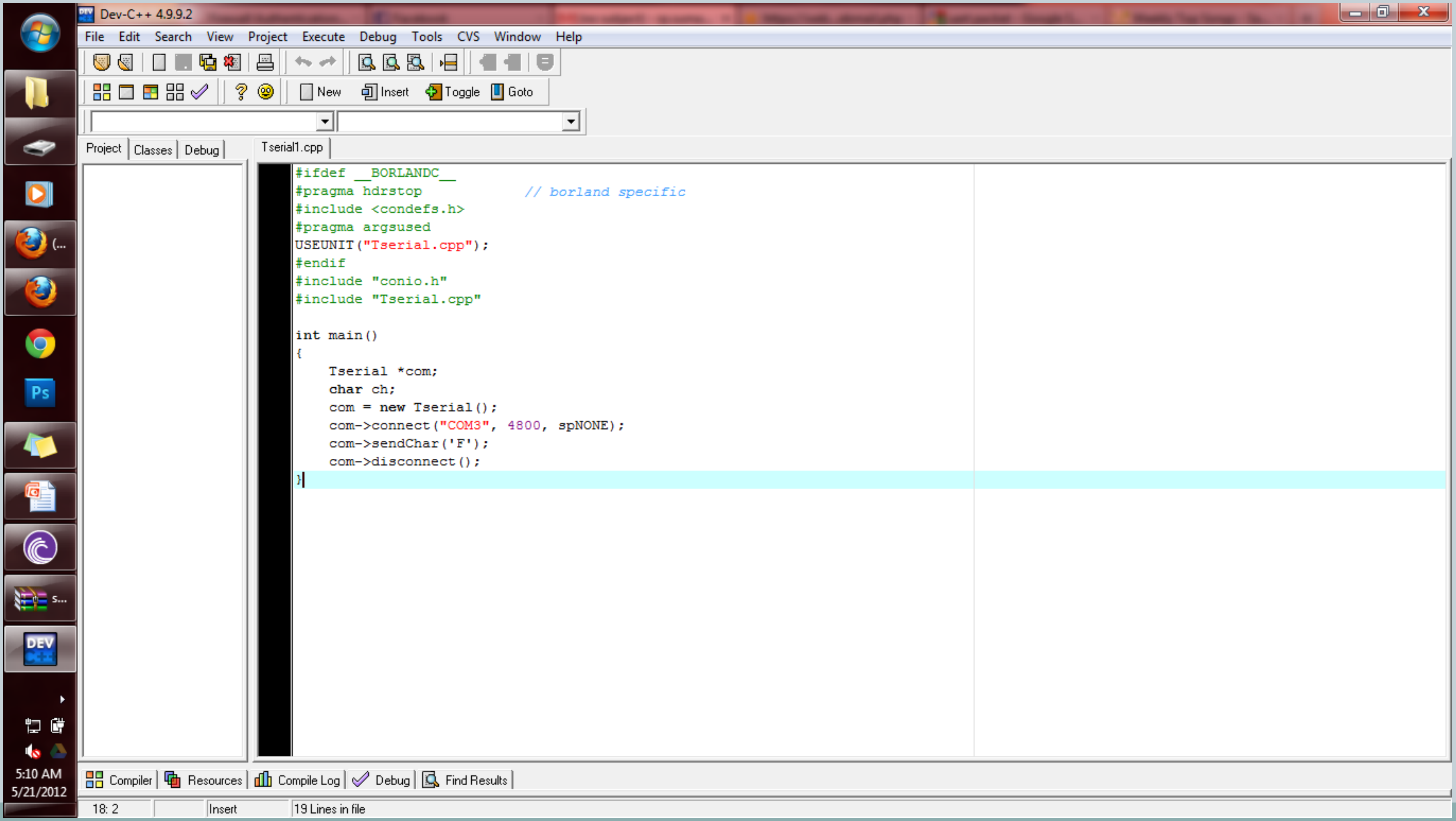
Coding in DevCPP



Coding in DevCPP



Coding in DevCPP


Dev-C++ 4.9.9.2
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
Project Classes Debug
Tserial1.cpp
Compiler Resources Compile Log Debug Find Results
18: 2
Insert
19 Lines in file
5:10 AM
5/21/2012

```
#ifdef __BORLANDC__
#pragma hdrstop            // borland specific
#include <condefs.h>
#pragma argsused
USEUNIT("Tserial.cpp");
#endif
#include "conio.h"
#include "Tserial.cpp"

int main()
{
    Tserial *com;
    char ch;
    com = new Tserial();
    com->connect("COM3", 4800, spNONE);
    com->sendChar('F');
    com->disconnect();
}
```

Coding in DevCPP

```
#ifdef __BORLANDC__
#pragma hdrstop          // borland specific
#include <condefs.h>
#pragma argsused
USEUNIT("Tserial.cpp");
#endif
#include "conio.h"
#include "Tserial.cpp"

int main(){
    Tserial *com;
    com = new Tserial();
    com->connect("COM3", 4800, spNONE);
    com->sendChar('F');
    com->disconnect();
}
```

Coding in DevCPP

```
#ifdef __BORLANDC__
#pragma hdrstop          // borland specific
#include <condefs.h>
#pragma argsused
USEUNIT("Tserial.cpp");
#endif
#include "conio.h"
#include "Tserial.cpp"

int main(){
   Tserial *com;
   com = new Tserial();
   com->connect("COM3", 4800, spNONE);
   com->sendChar('F');
   com->disconnect();
}
```

$\longrightarrow$ For Including "Tserial.cpp" library.place "Tserial.Cpp " with your code just place it in same folder where your code is presnt

Coding in DevCPP

```
#ifdef __BORLANDC__
#pragma hdrstop          // borland specific
#include <condefs.h>
#pragma argsused
USEUNIT("Tserial.cpp");
#endif
#include "conio.h"
#include "Tserial.cpp"

int main(){
   Tserial *com;
   com = new Tserial();
   com->connect("COM3", 4800, spNONE);
   com->sendChar('F');
   com->disconnect();
}
```

$\boxed{\texttt{Tserial *com;}} \longrightarrow$ Object Declaration

Coding in DevCPP

```
#ifdef __BORLANDC__
#pragma hdrstop            // borland specific
#include <condefs.h>
#pragma argsused
USEUNIT("Tserial.cpp");
#endif
#include "conio.h"
#include "Tserial.cpp"

int main(){
    Tserial *com;
    com = new Tserial();
    com->connect("COM3", 4800, spNONE);
    com->sendChar('F');
    com->disconnect();
}
```

$\xrightarrow{\hspace{6em}}$ Object Creation

Coding in DevCPP

```
#ifdef __BORLANDC__
#pragma hdrstop          // borland specific
#include <condefs.h>
#pragma argsused
USEUNIT("Tserial.cpp");
#endif
#include "conio.h"
#include "Tserial.cpp"

int main(){
   Tserial *com;
   com = new Tserial();
   com->connect("COM3", 4800, spNONE);
   com->sendChar('F');
   com->disconnect();
}
```

$\longrightarrow$ Connecting to a serial port

Coding in DevCPP

```
#ifdef __BORLANDC__
#pragma hdrstop          // borland specific
#include <condefs.h>
#pragma argsused
USEUNIT("Tserial.cpp");
#endif
#include "conio.h"
#include "Tserial.cpp"

int main(){
   Tserial *com;
   com = new Tserial();
   com->connect("COM3", 4800, spNONE);
   com->sendChar('F');
   com->disconnect();
}
```

$\xrightarrow{}$ Send Character on Com port

Coding in DevCPP

```
#ifdef __BORLANDC__
#pragma hdrstop          // borland specific
#include <condefs.h>
#pragma argsused
USEUNIT("Tserial.cpp");
#endif
#include "conio.h"
#include "Tserial.cpp"

int main(){
    Tserial *com;
    com = new Tserial();
    com->connect("COM3", 4800, spNONE);
    com->sendChar('F');
    com->disconnect();
}
```

$\xrightarrow{\hspace{3em}}$ Don’t forget to disconnect Com port

Coding in DevCPP

```
#ifdef __BORLANDC__
#pragma hdrstop          // borland specific
#include <condefs.h>
#pragma argsused
USEUNIT("Tserial.cpp");
#endif
#include "conio.h"
#include "Tserial.cpp"

int main(){
    Tserial *com;
    com = new Tserial();
    com->connect("COM3", 4800, spNONE);
    com->sendChar('F');
    com->disconnect();
}
```

ELECTRONICS CLUB IITKANPUR

Coding in DevCPP

For More Details

http://www.tetraedre.com/advanced/serial/index.html

Opening Com Port

Python

Matlab

JAVA

C Lang

Thank You
Question??