Wireless Communication Through Serial Port

How to interface computer’s serial port with a microcontroller

Animesh Tyagi
B.Tech IT
AITM,Kanpur
1471913002

What is a Serial Port exactly ?

In computing, a **serial port** is a serial communication interface through which information transfers in or out one bit at a time (in contrast to a parallel port).Throughout most of the history of personal computers, data was transferred through serial ports to devices such as modems, terminals, and various peripherals.

While such interfaces as Ethernet, FireWire, and USB all send data as a serial stream, the term "serial port" usually identifies hardware more or less compliant to the RS-232 standard, intended to interface with a modem or with a similar communication device.

OH! So a to and fro communication port like USB so why not go along with the USB ?

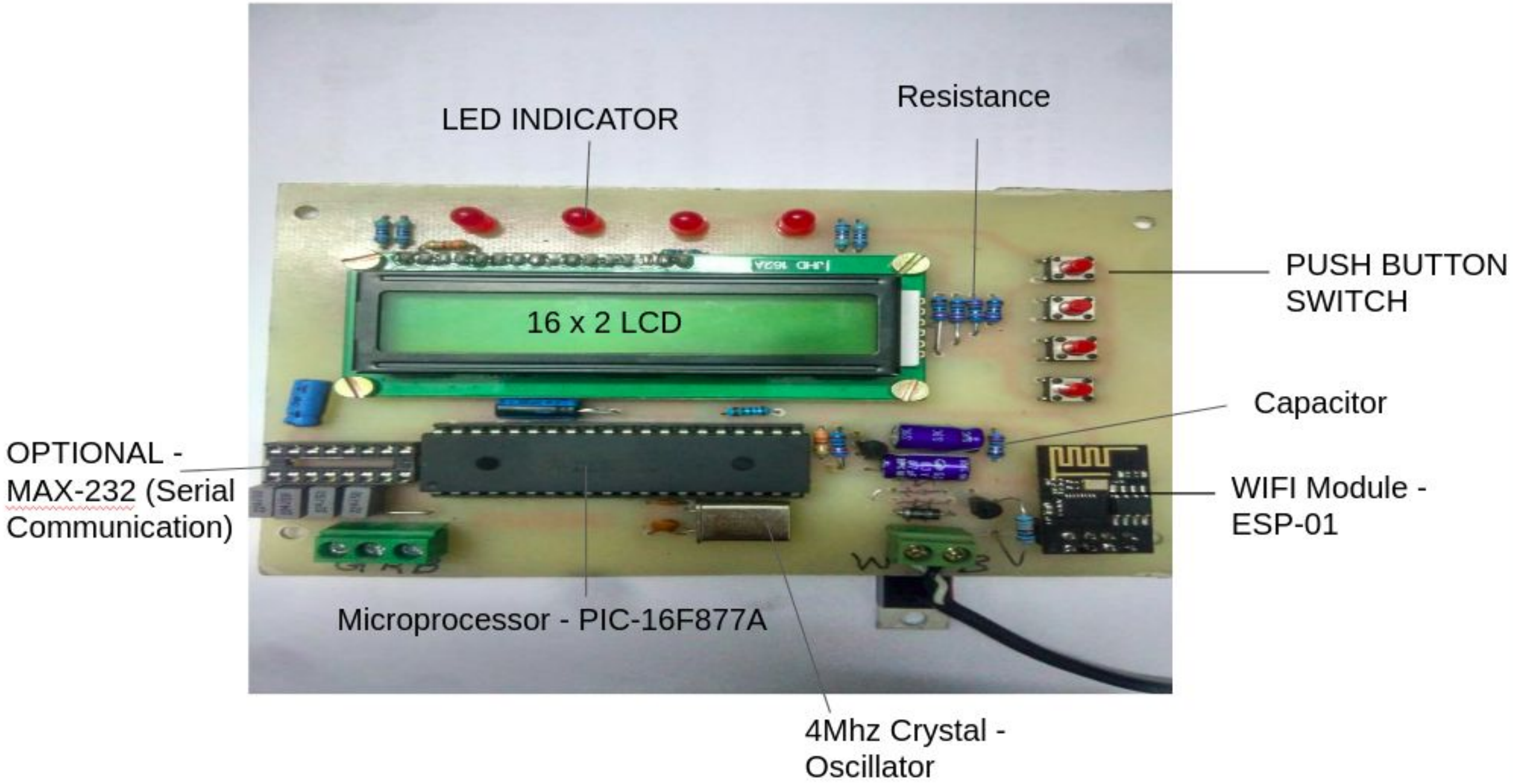
To answer your question we can go along with the USB and in most areas we do, but the only reason that serial ports are still relevant in on Windows these days is because a USB device requires a custom device driver. Device manufacturers do *not* like writing and supporting drivers, they often take a shortcut in their driver that makes it emulate a legacy serial port device. So programmers can use the legacy support for serial ports built into the operating system and about any language runtime library.

USB TTL Module

Most commonly the USB data signals are converted to either RS232, RS485, RS422, or TTL-level UART serial data.

The USB TTL Serial cables are a range of USB to serial converter cables which provide connectivity between USB and serial UART interfaces

SO NOW TO LOOK AT THE HARDWARE WE’RE WORKING WITH IN THIS PROJECT


LED INDICATOR
Resistance
PUSH BUTTON SWITCH
16 x 2 LCD
Capacitor
OPTIONAL - MAX-232 (Serial Communication)
WIFI Module - ESP-01
Microprocessor - PIC-16F877A
4Mhz Crystal - Oscillator

Hardware Requirements

- Power supply of DC 5V (provided through attached charger)
- A Computer with wireless communication enabled to configure Circuit board through HyperTerminal
- Oracle Java 8 or latest installed on the computer with atleast 512MB RAM free to allocate to the program
- The HyperTerminal Program installed on the computer & running to communicate with the board

Major Contributing Hardware in Circuit Board -

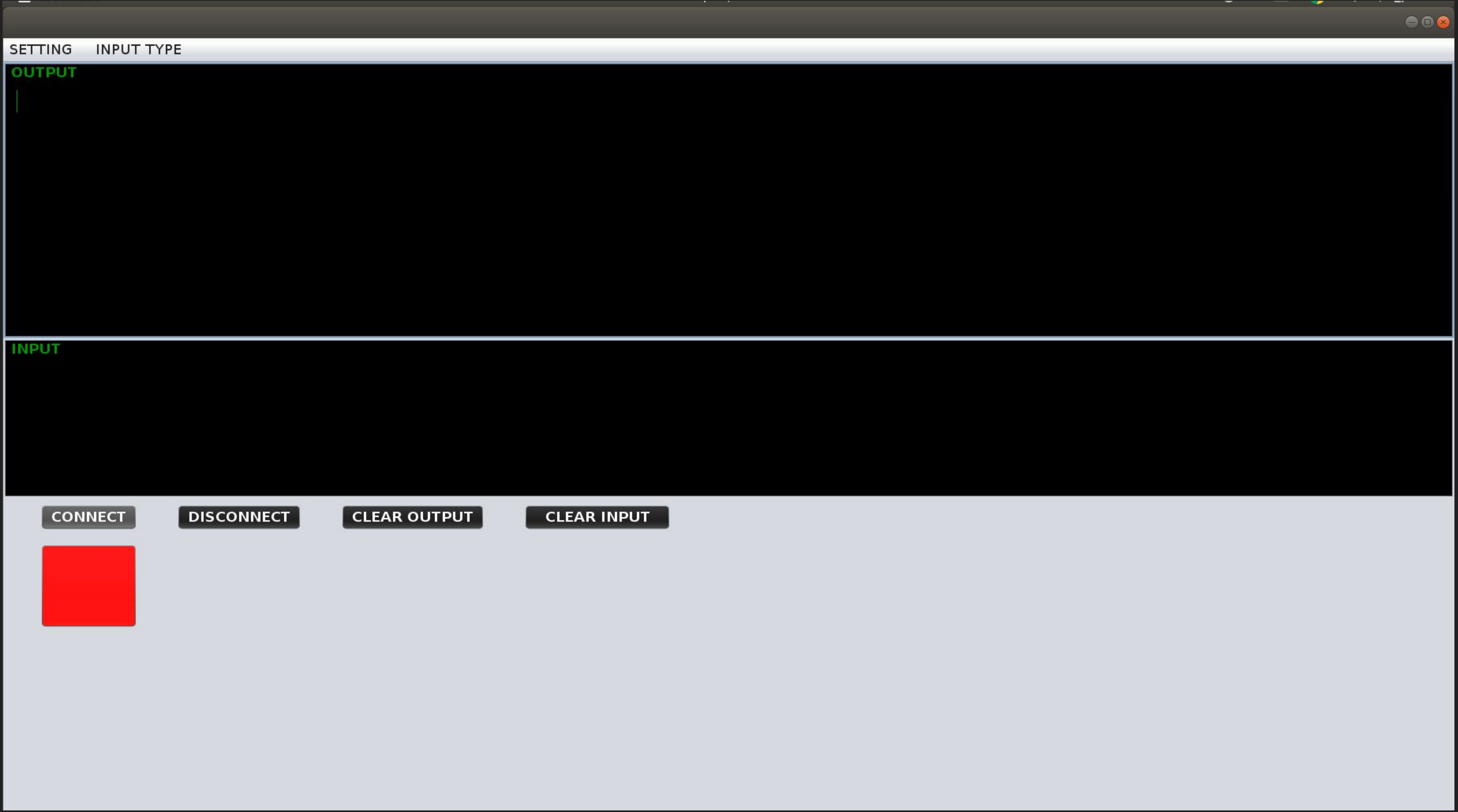
Microprocessor - PIC-16F877A -

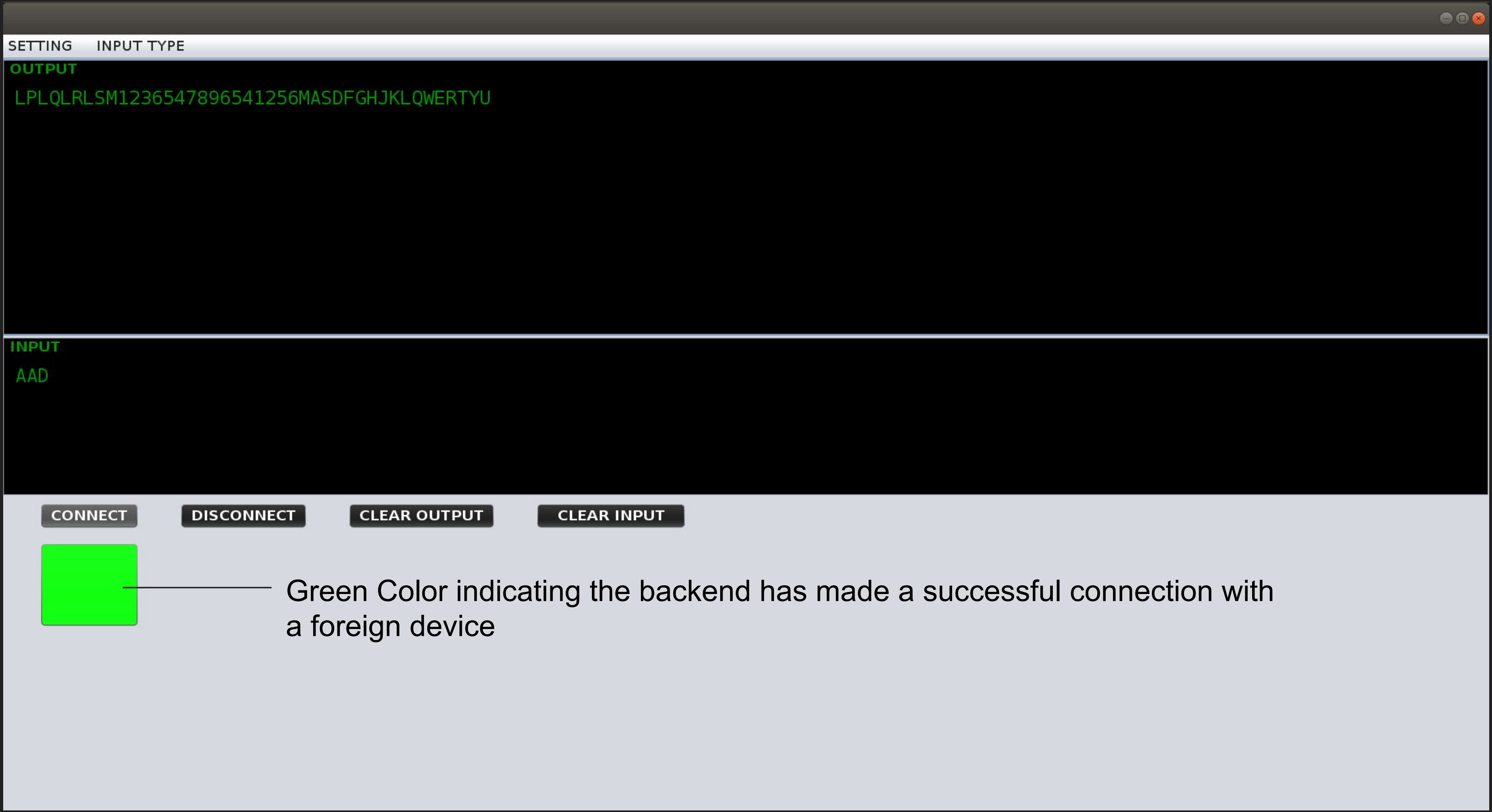
2 PWM 10-bit 256 Bytes EEPROM data memory ICD 25mA sink/source per I/O Self Programming Parallel Slave Port

WIFI Module - ESP-01 -

ESP-01 is a highly integrated chip designed for the needs of a new connected world. It offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor.

NOW A LOOK AT THE COMPUTER END IN A HYPERTERMINAL - “GUI_PLAY.jar”


SETTING
INPUT TYPE
OUTPUT
INPUT
CONNECT
DISCONNECT
CLEAR OUTPUT
CLEAR INPUT



OUTPUT

```
LPLQLRLSM1236547896541256MASDFGHJKLQWERTYU
```

INPUT

```
AAD
```

Green Color indicating the backend has made a successful connection with a foreign device

Now to see the actual Functions we can do with this successful connection

TO SHOW INPUT RECEIVED FROM MICROCONTROLLER

Button 1 = A
Button 2 = B
Button 3 = C
Button 4 = D

TO SHOW OUTPUT FROM COMPUTER TO MICROCONTROLLER

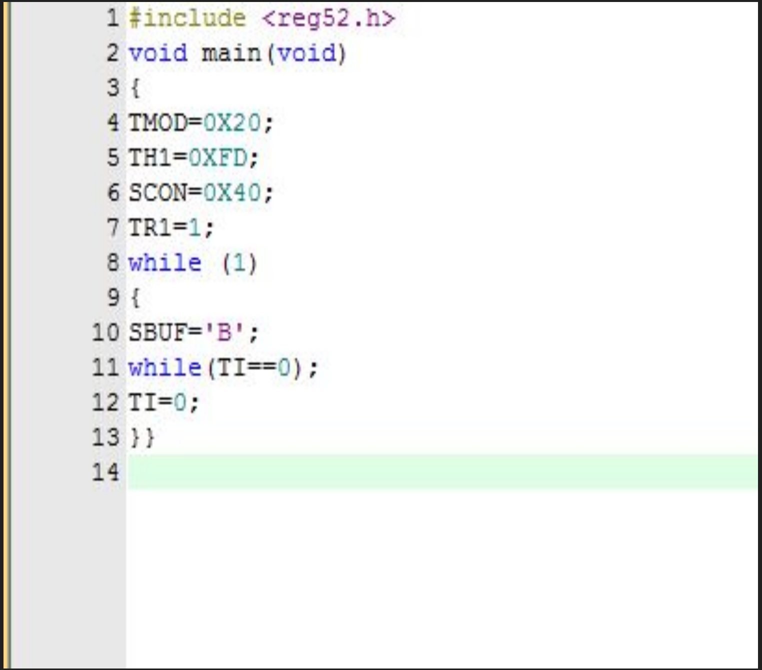
To Turn on LED 1 = LP
To Turn on LED 2 = LQ
To Turn on LED 3 = LR
To Turn on LED 4 = LS

TO DISPLAY A MESSAGE FROM COMPUTER TO MICROCONTROLLER

M<Followed by 16 Alphanumeric Characters including spaces>

Now let’s take a look at the programming end for MAX-RS232

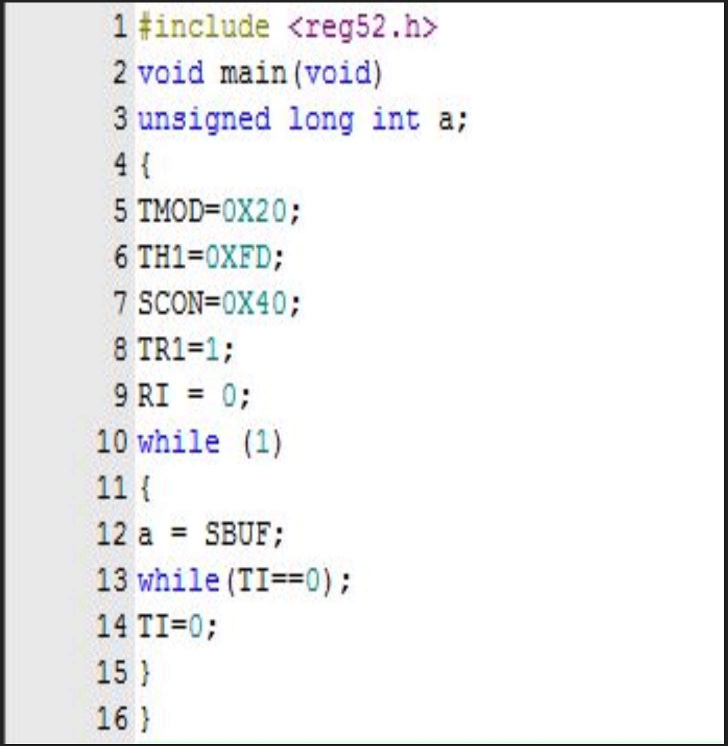
1. Serial Transmission



```
#include <reg52.h>
void main(void)
{
TMOD=0X20;
TH1=0XFD;
SCON=0X40;
TR1=1;
while (1)
{
SBUF='B';
while(TI==0);
TI=0;
}}
```

Program to transmit the character 'B' serially

2. Serial Reception



```c
#include <reg52.h>
void main(void)
unsigned long int a;
{
TMOD=0X20;
TH1=0XFD;
SCON=0X40;
TR1=1;
RI = 0;
while (1)
{
a = SBUF;
while(TI==0);
TI=0;
}
}
```

Program to receive the character 'B' serially

So by now you must be wondering how useful is this project as it just shows a bidirectional flow of communication

So here are some examples where serial port is used extensively :-

ARDUINO

Dial-up modems

Configuration and management of networking equipment such as routers, switches, firewalls, load balancers

GPS receivers (typically NMEA 0183 at 4,800 bit/s)

Bar code scanners and other point of sale devices

LED and LCD text displays

Satellite phones, low-speed satellite modems and other satellite based transceiver devices

Flat-screen (LCD and Plasma) monitors to control screen functions by external computer, other AV components or remotes

Test and measuring equipment such as digital multimeters and weighing systems

Updating firmware on various consumer devices.

CNC controllers

Uninterruptible power supply

Stenography or Stenotype machines.

Software debuggers that run on a second computer.

Industrial field buses

Printers

Computer terminal, teletype

Older digital cameras

Networking (Macintosh AppleTalk using RS-422 at 230.4 kbit/s)

Serial mouse

Arduino MIcrocontroller Boards

So now that we can control a Micro Controller from our computer with the help of our HyperTerminal, it’s a matter of few changes in the backend to control some other SerialComm device

So the Possibilities of this project are endless