8 Bit Data Transfer By Parallel Port Connection

Firas Shawkat Hamid, Ahmad M. Alsheikh, Kudama A. Hamid
Electronic Department, Technical Institute in Mosul, Mosol, Iraq

Abstract

This paper presents a design and performing a special protocol. Using outlet parallel port from EPP's kind does not used before [1,2] for the operation of data transmission between two computers using 8 bits data transmission.

A program was designed using especial protocol for data transmission from and to the computer without dependence on any of the used protocols in the world [1,2,3]with the least possible error. This program was designed using Qbasic language.

This program consist of several functions such as window function to indicate whether the computer in reception or transmission or exit case, transmission function, reception function and a function to open the file to be send, bits check function on parallel port. This enables us to determine the transmission and reception time and its status.

This program work after providing data cable, which is, depends according to established program.

Introduction

When IBM introduced the PC, in 1981, the parallel printer port was included as an alternative to the slower serial port as a means for driving the latest high performance dot matrix printers [1,2]. The parallel port had the capability to transfer 8 bits of data at a time where as the serial port transmitted one bit at a time. when the PC was introduced, dot matrix printers were the main peripheral that used the parallel port. As technology progressed and the need for greater external connectivity increased, the parallel port became the means by which you could connect higher performance peripherals. The peripherals now range from printer sharing devices, portable disk drivers and tape backup to local area network adapters and CDROM players. The data lines are used to provide data from PC to the printer in that direction only. Later implementations of the parallel port allowed for data to be driven from the peripherals to the PC. Table (1) identifies each of these signals and gives their Standard Parallel Port (SPP) definitions [1]. The signals within these groups are assigned to specific bits within the registers that make up the hardware/software interface to the parallel port .The parallel port is mapped in to the I/O so ace of the PC[1].

Parallel Port Groups

The parallel port, as implemented on the PC, consists of a connector with 17 signal lines and 8 ground lines [2,3]. The signal lines are divided into three groups:

Control (4 lines), Status (5 lines), Data (8 lines).

As originally designed the control lines are used as interface control and handshaking signal from the PC to the printer. The status lines are used for handshake signals and the status indicators for such mode. Many of these signals are used for mode transitions and for additional status information [1,3].

The EPP (Enhanced Parallel Port) and ECP (Extended Capabilities Port)modes have Bi-directional capability as part of there protocol. These modes require that the Hardware implement a state machine that is capable of automatically generating the control strobes that are necessary for these high performance data transfer modes [1,2].

The Parallel Port is the most commonly used port for interfacing home made projects. This port will allow the input of up to 9 bits or the output of 12 bits at any one given time, thus requiring minimal external circuitry to implement many simpler tasks. The port is composed of 4 control lines,5 status lines and 8 data lines. It's found commonly on the back of your PC as a D-Type 25 Pin female connector. There may also be a D-Type 25 pin male connector. This will be a serial RS-232 port and thus, is a totally incompatible port. Newer Parallel Port's are standardized under the IEEE 1284 standard first released in 1994. This standard defines 5 modes of operation which are as follows[1,3],

1. Compatibility Mode.
2. Nibble Mode. *(Protocol not Described in this Document)*
3. Byte Mode. *(Protocol not Described in this Document)*
4. EPP Mode *(Enhanced Parallel Port).*
5. ECP Mode *(Extended Capabilities Port).*

The aim was to design new drivers and devices which were compatible with each other and Interfacing the Standard Parallel Port.

Parallel Port Connection

Compatibility, Nibble & Byte modes use just the standard hardware available on the original Parallel Port cards while EPP & ECP modes require additional hardware which can run at faster speeds, while still being downwards compatible with the Standard Parallel Port[1].

Compatibility mode or "Centronics Mode" as it is commonly known, can only send data in theforward direction at a typical speed of 50 kbytes per second but can be as high as 150+ kbytes asecond. In order to receive data, you must change the mode to either Nibble or Byte mode. Nibble mode can input a nibble *(4 bits)* in the reverse direction. E.g. from device to computer. Byte mode uses the Parallel's bi-directional feature (found only on some cards) to input a byte *(8 bits)* of data in the reverse direction[1,4].

Extended and Enhanced Parallel Ports use additional hardware to generate and manage handshaking. To output a byte to a printer (or anything in that matter) using compatibility mode, the software must,**1**-Write the byte to the Data Port. **2-**Check to see if the printer is busy. If the printer is busy, it will not accept any data, thus any data which is written will be lost. **3-**Take the

Strobe (Pin 1) low. This tells the printer that there is the correct data on the data lines (Pins 2-9). **4-**Put the strobe high again after waiting approximately 5 microseconds after putting the strobe low (Step 3)[1,3].

This limits the speed at which the port can run at. The EPP & ECP ports get around this by letting the hardware check to see if the printer is busy and generate a strobe and /or appropriate handshaking. This means only one I/O instruction need to be performed, thus increasing the speed.

These ports can output at around 1-2 megabytes per second. The ECP port also has the advantage of using DMA (Direct Memory Access) channels and FIFO (First In First Out) buffers, thus data can be shifted around without using I/O instructions[3].

Hardware Properties

The output of the Parallel Port is normally TTL(Transistor Transistor Logic) logic levels. The voltage levels are the easy part. The current you can sink and source varies from port to port. Most Parallel Ports implemented in ASIC, can sink and source around 12mA. However these are just some of the figures taken from Data sheets, Sink/Source 6mA, Source 12mA/Sink 20mA, Sink 16mA/Source 4mA, Sink/Source 12mA. As you can see they vary quite a bit. The best bet is to use a buffer, so the least current is drawn from the Parallel Port[4].

Signals

The following table (1) shows the pins layout[1]:

Table (1) Pins Layout Connector: female DB25

Pins	Name	Description	D		Pins	Name	Description	D
1	STROBE	Strobe	- I/O		14	AUTOFEED	Auto-Feed	- O
2	D0	Data Bit 0	I/O		15	ERROR	Error	- I
3	D1	Data Bit 1	I/O		16	INIT	Init (Reset)	- O
4	D2	Data Bit 2	I/O		17	SLCT IN	Select in	- O
5	D3	Data Bit 3	I/O		18	GND	Ground	/
6	D4	Data Bit 4	I/O		19	GND	Ground	/
7	D5	Data Bit 5	I/O		20	GND	Ground	/
8	D6	Data Bit 6	I/O		21	GND	Ground	/
9	D7	Data Bit 7	I/O		22	GND	Ground	/
10	ACK	Acknowledge	- I		23	GND	Ground	/
11	BUSY	Busy	I		24	GND	Ground	/
12	PE	Paper End	I		25	GND	Ground	/
13	SLCT	Select out	I		S	GND	Chasis Ground	/

STROBE (Strobe) active low output Notify the printer that data available on D0 to D7 are Valid.

D0 – D7 (Data Bus) Outputs data bytes send to the printer output only in "Compatible" mode and Bi-directional in newer modes.

ACK (Acknowledge) active low input Notify the computer that the printer is ready to receive the next data.

BUSY (Busy) active low input Printer Buffer full or printer Busy, the computer must wait for this signal to get high again to continue sending data.

PE (Paper End) active high input Printer out of paper.

SLCT (Select Out) active high input Printer readies (On-Line).

AUTO-FEED (Auto-Feed) active high output Printer line feed.

ERROR (Error) active low input Error detected by the printer.

INIT (Reset) active low output Initialize the printer (Reset).

SLCT IN (Select In) active low output Send an on-line request to the printer.

PPM and the ECP's Extended Control Register

The ECP registers are standardized under Microsoft's Extended Capabilities port protocol and ISA interface standard, thus we don't have problem of every vendor having there own register set[3].

When set to ECP mode, a new set of registers become available at base + 0X402H. a discussion of these registers are available in interfacing the extended capabilities port. Here we are only interested in the Extended Control Register (ECR) which is mapped at Base + 0X402H.it should be stated that the ECP's registers are not available for port's with abase address of 0X3BCh[2,4].

Table (2) PPM (Parallel Port Modes)

Bit	Function	
7:5	Select Current Mode of Operation	
	000	Standard Mode
	001	Byte Mode
	010	Parallel Port FIFO Mode
	011	ECP FIFO Mode
	100	EPP Mode
	101	Reserved
	110	FIFO Test Mode
	111	Configuration Mode
4	ECP Interrupt Bit	
3	DMA Enable Bit	
2	ECP Service Bit	
1	FIFO Full	
0	FIFO Empty	

The table (2) above is of the extended control register. We are only interested in the three MSB of the extended control register, which select the mode of operation. There are 7 possible modes of operation, but not all ports will support all modes. The EPP mode is one such example, not being available on some ports.

The non bi-directional ports were manufactured with the 74LS374's output enable tied permanent low, thus the data port is always output only. When you read the parallel port's data register, the data comes from.

EPP Connection Software & Hardware Design

Each signal is identified by it's pin number on a DB-25 connector and its signal name, as shown in table (3)[1]:

Table (3) EPP (Enhanced Parallel Port)

Pin No.	Signal Name
0	Write #
1	Address / Data 0
2	Address / Data 0
3	Address / Data 0
4	Address / Data 0
5	Address / Data 0
6	Address / Data 0
7	Address / Data 0
8	Address / Data 0
9	Address / Data 0
10	Interrupt #
11	Wait # (Paired with 16)
12	Paper End
13	Select
14	Data Strobe #
15	Error
16	Initialize Printer (Paired with 11)
17	Address Strobe #
18	Ground (Data)
19	Ground (Paired with 1)
20	Ground (Paired with 10)
21	Ground (Paired with 12)
22	Ground (Paired with 13)
23	Ground (Paired with 14)
24	Ground (Paired with 15)
25	Ground (Paired with 17)

This Port can output at around 1-2 Megabytes Per Second. The ECP Port also has the advantage of using DMA channels and FIFO buffers, thus data can shifted around without using I/O instructions.

Software Registers Parallel Port

Table (4) Data Port

Offset	Name	Read/Write	Bit No.	Properties
Base + 0	Data Port	Write	Bit 7	Data 7
			Bit 6	Data 6
			Bit 5	Data 5
			Bit 4	Data 4
			Bit 3	Data 3
			Bit 2	Data 2
			Bit 1	Data 1
			Bit 0	Data 0

The base address, usually called the data port or data register as shown in table (4) is simply used for outputting data on the parallel port's data lines (Pins 2-9). This register is normally a write only port. If you read from the port, you should get the last byte sent. However if your port is Bi-directional, you can receive data on this address[1,3].

Table (5) Status Port

Offset	Name	Read/Write	Bit No.	Properties
Base + 1	Status Port	Read Only	Bit 7	Busy
			Bit 6	Ack
			Bit 5	Paper Out
			Bit 4	Select in
			Bit 3	Error
			Bit 2	IRQ (Not)
			Bit 1	Reserved
			Bit 0	Reserved

The status port as shown in table (5) (base address + 1) is a read only port. Any data written to this port will be ignored. The status port is made up of 5 input lines (Pins 10,11,12,13 & 15), an IRQ status register and two reserved bits. Please note that bit 7 (Busy) is an active low input[1,3].

Table (6) Control Port

Offset	Name	Read/Write	Bit No.	Properties
Base + 2	Control Port	Read/Write	Bit 7	Unused
			Bit 6	Unused
			Bit 5	Enable bi-directional Port
			Bit 4	Enable IRQ Via Ack Line
			Bit 3	Select printer
			Bit 2	Initialize Printer (reset)
			Bit 1	Auto Linefeed
			Bit 0	Strobe

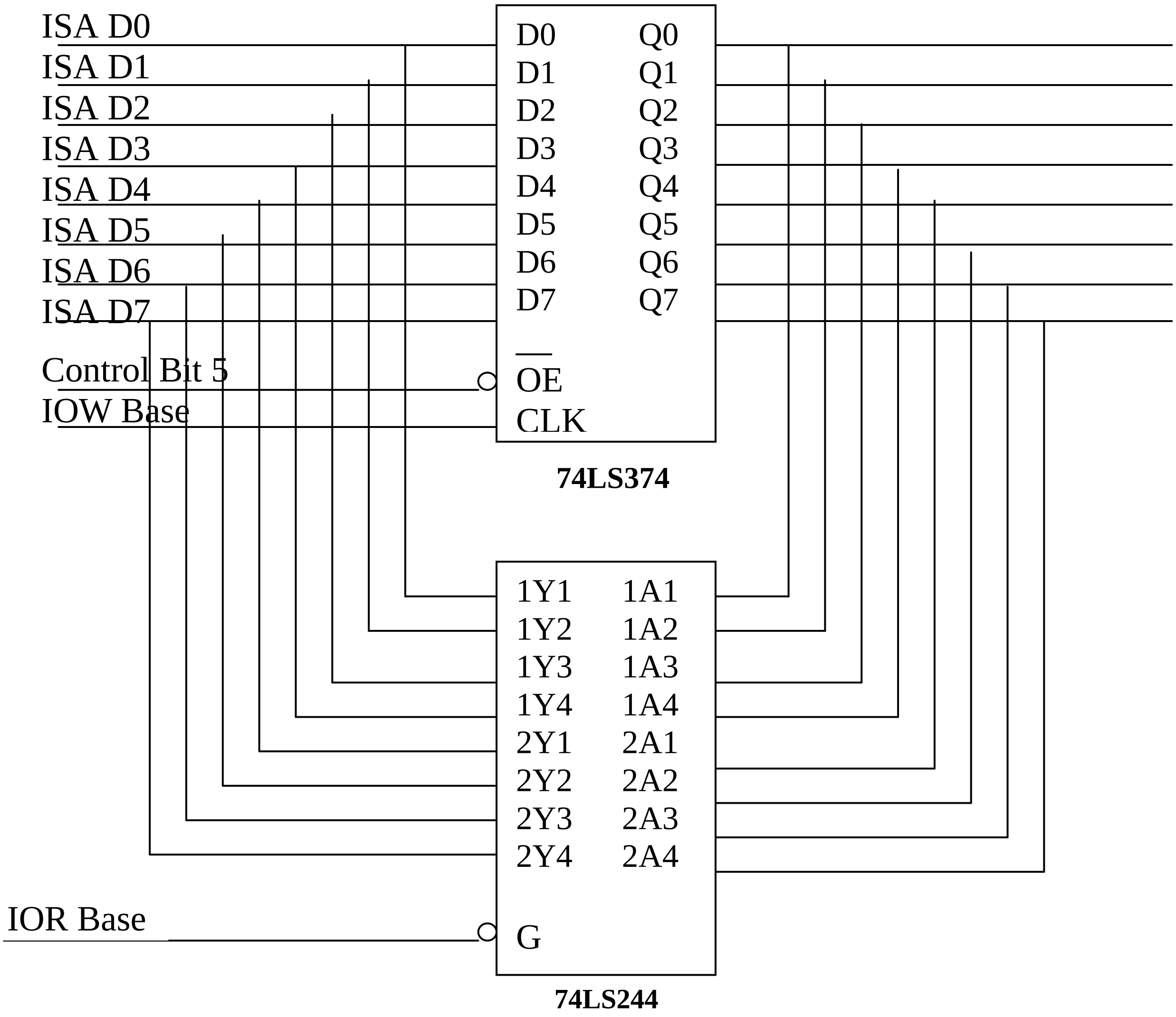
The control port as shown in table (6) (base address + 2) was intended as a write only port. When the printer is attached to the parallel port, for "controls" are used. These are strobe, Auto linefeed, initialize and select printer, all of which are inverted except initialize[1,3].

These four outputs can also be used for inputs. If the computer has placed a pin high (e.g. +5V) and your device wanted to take it low, you would effectively short out the port, causing a conflict on that pin. Therefore these lines are "Open Collector" outputs (or open drain for CMOS devices). This means that it has two states. A low state (0V) and high impedance state (open circuit)[4].

Bi-directional Ports

The schematic diagram below in figure (1) shows a simplified view of parallel port's data register. The original parallel port card's implemented 74LS logic[1,4].

Figure (1) Parallel Port bi-directional Operation



Experimental Work

A new cable connection has been established to realize a special protocol that mentioned before in the abstract , as shown in table (7) which gave a high speed transmission and a good data that has a negligible error:

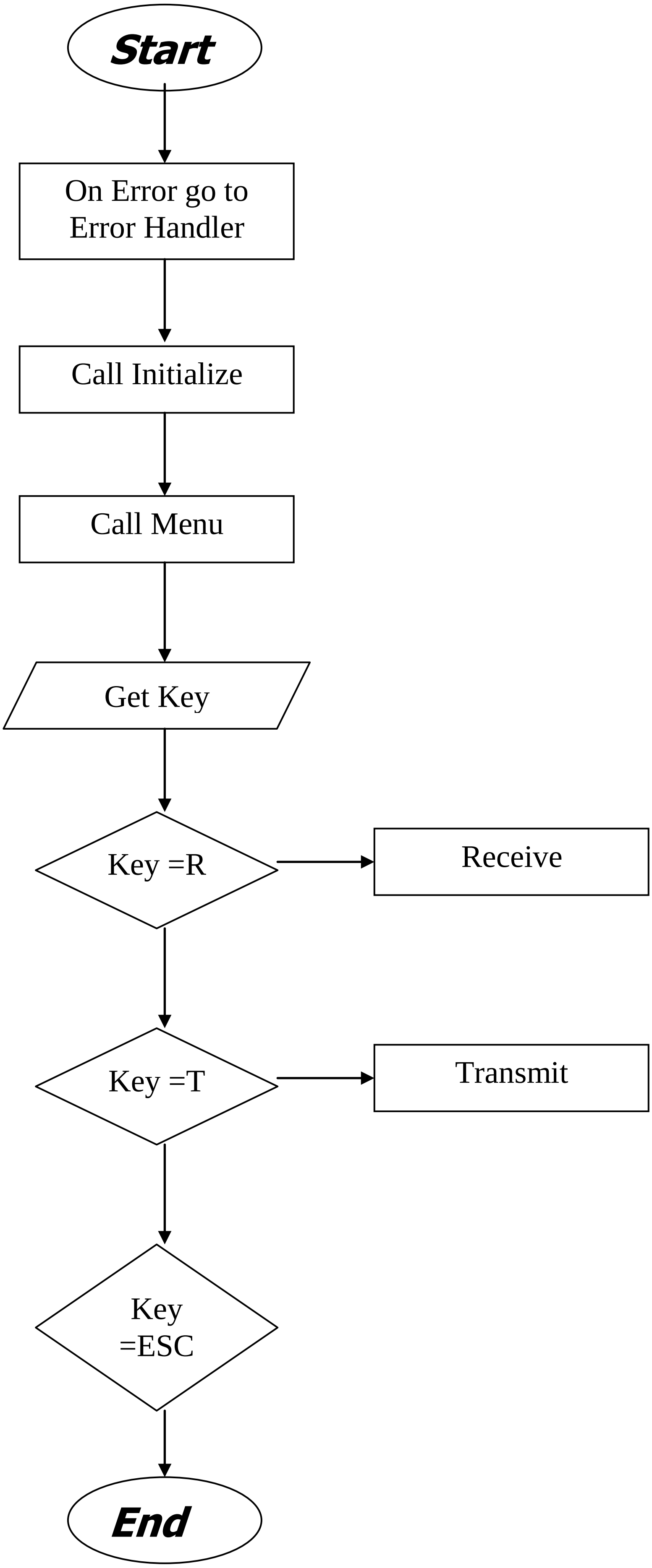
Table (7) Cable Connection

Pin No. for PC1	Signal Name	Pin No. for PC2
2	Address / Data 0	2
3	Address / Data 0	3
4	Address / Data 0	4
5	Address / Data 0	5
6	Address / Data 0	6
7	Address / Data 0	7
8	Address / Data 0	8
9	Address / Data 0	9
1	Strobe → ACK	10
10	ACK ← Strobe	1
11	Busy ← INIT	16
16	INIT → Busy	11
13	Select out ← Select in	17
17	Select in → Select out	13
25	Ground	25

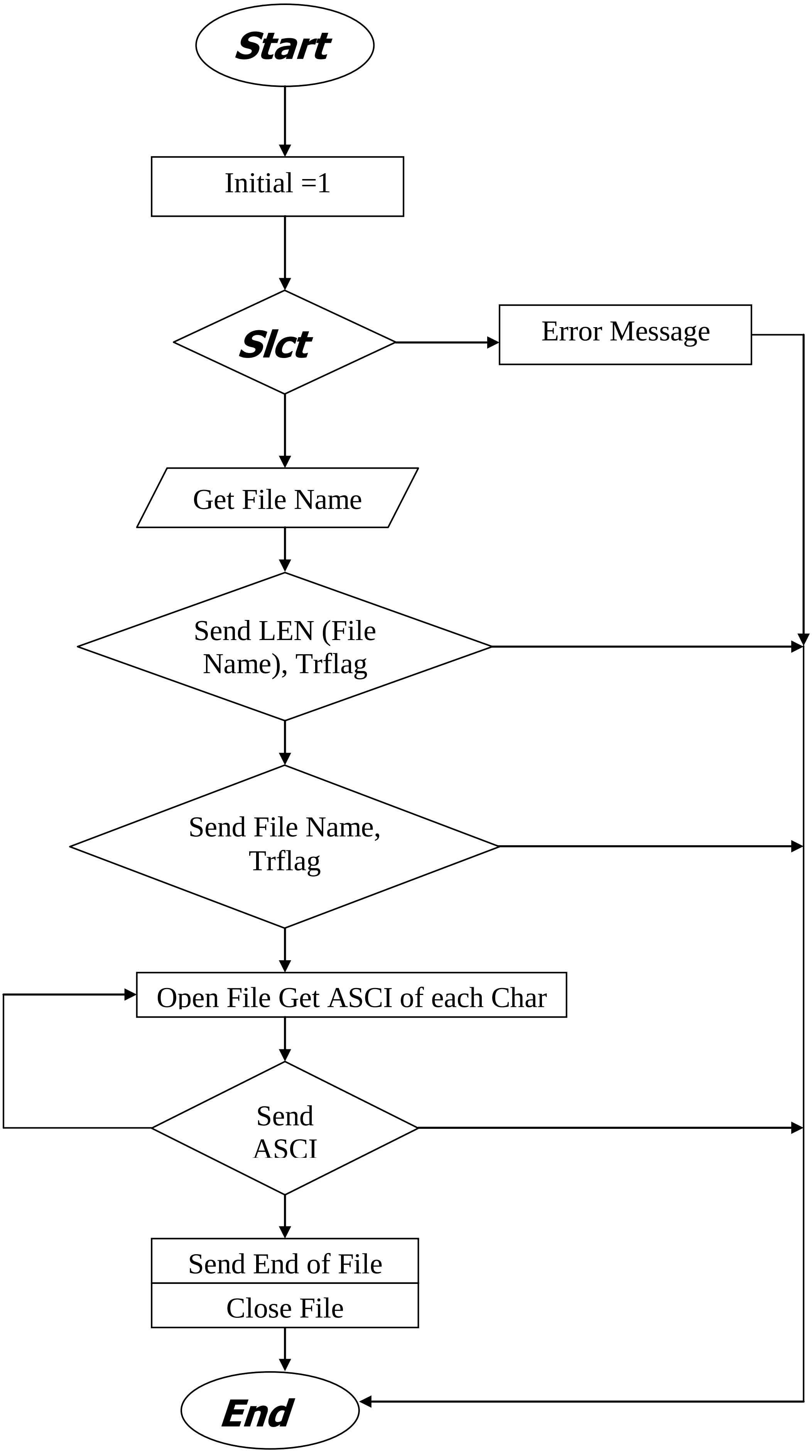
Flow chart

Now we explain the flowchart for the program that executes to transmit and receive 8 bit using EPP:

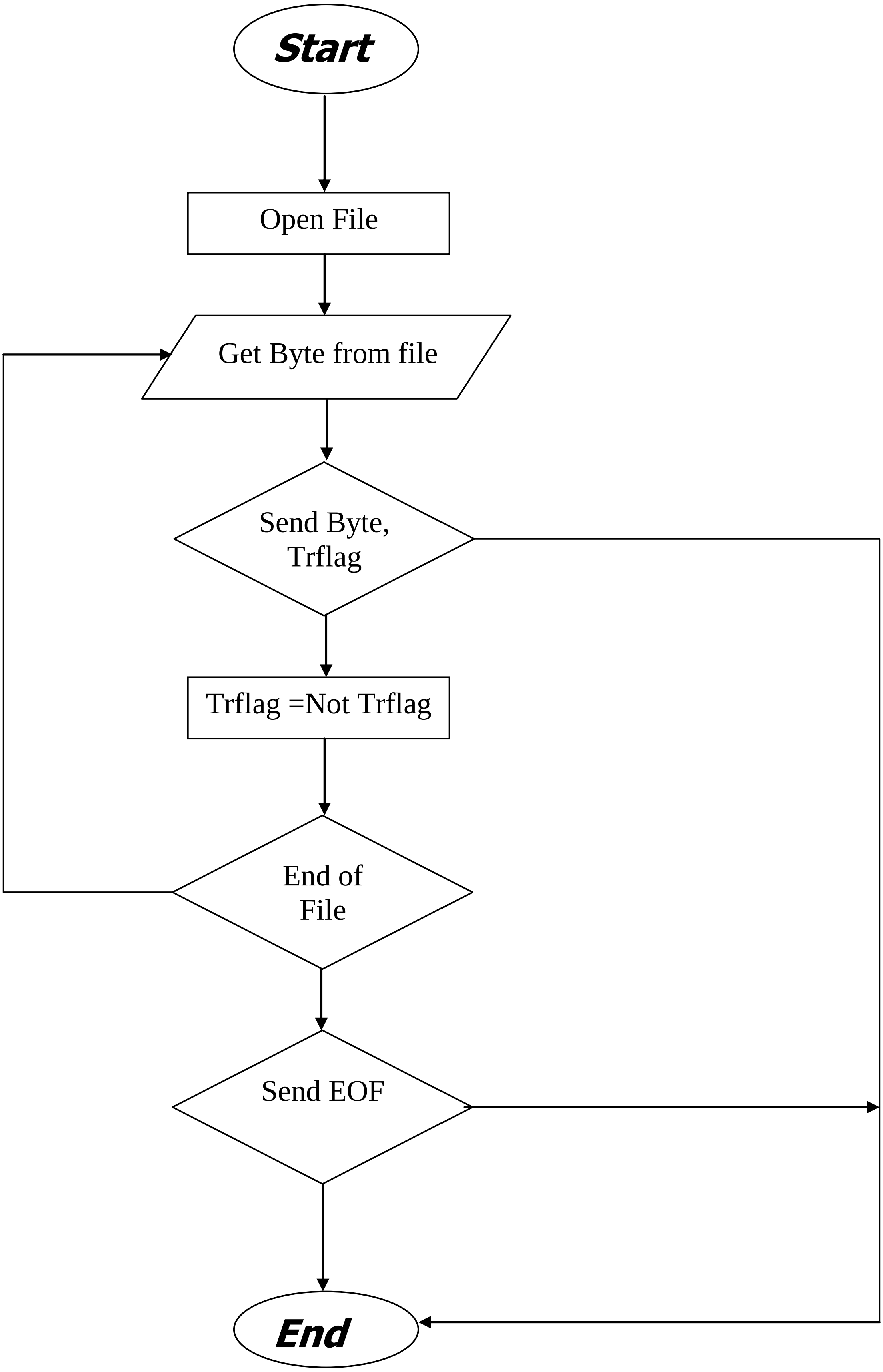
Main program



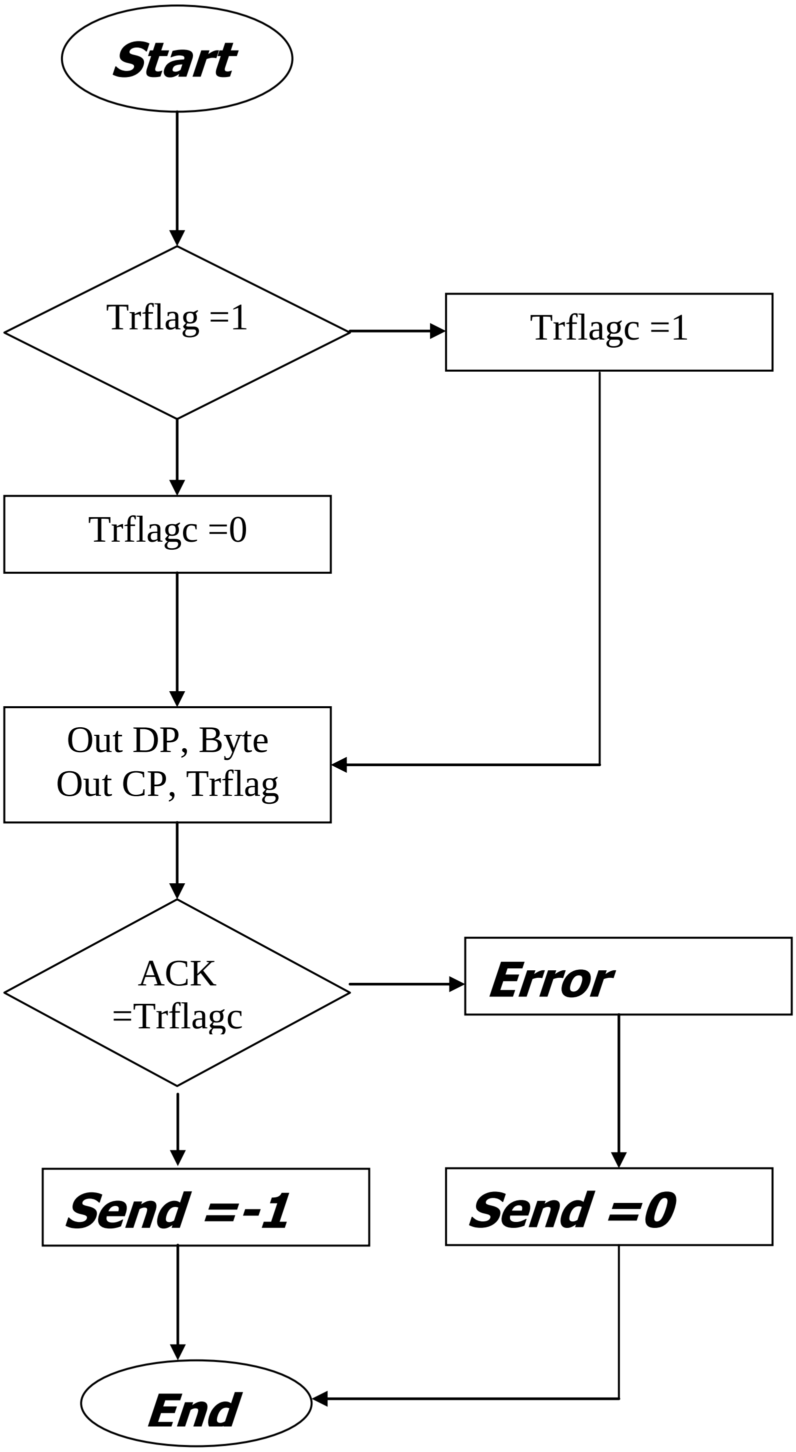
Transmit


Start
Initial =1
Slct
Error Message
Get File Name
Send LEN (File Name), Trflag
Send File Name, Trflag
Open File Get ASCI of each Char
Send ASCI
Send End of File
Close File
End

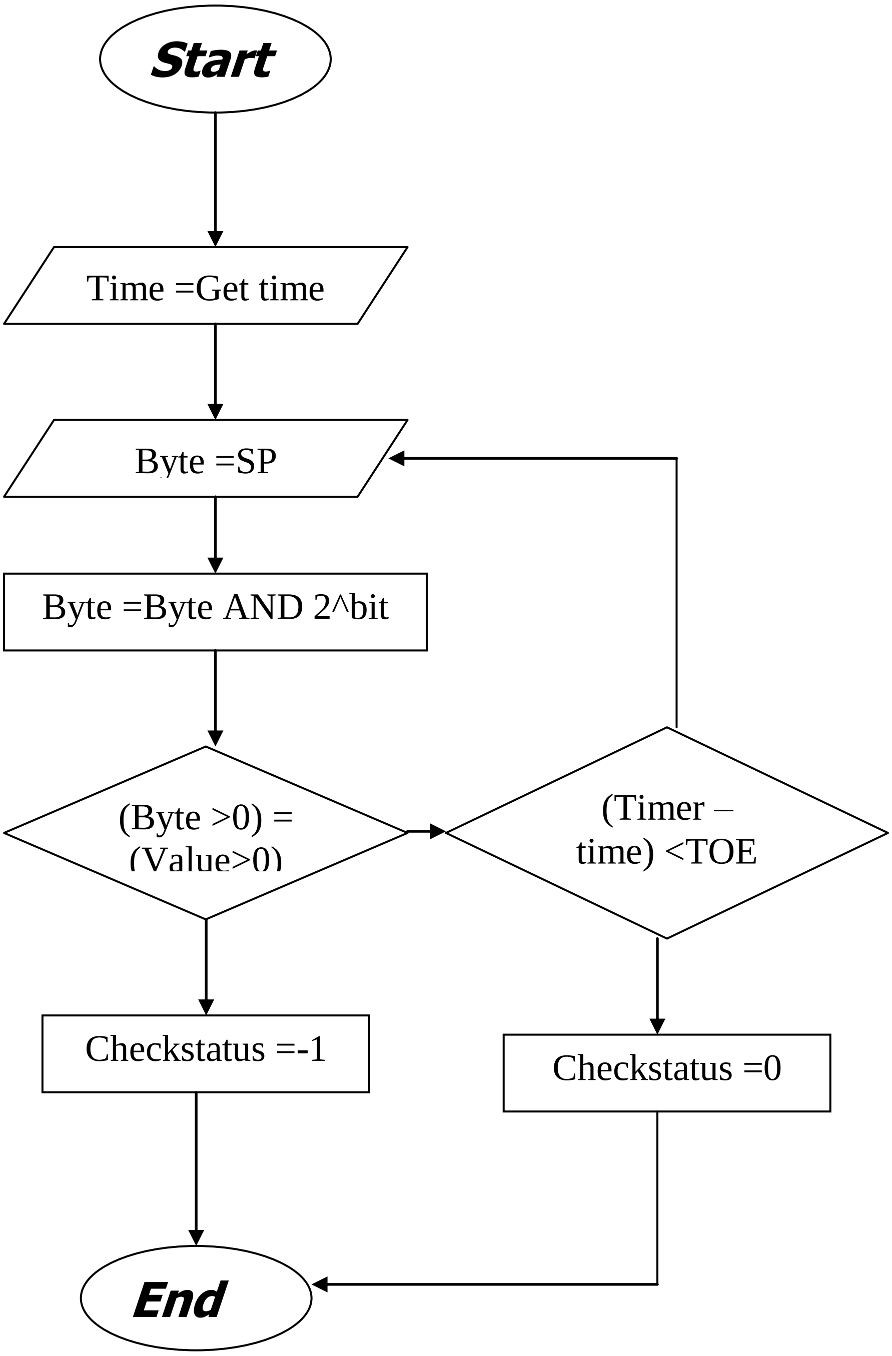
Open File


Start
Open File
Get Byte from file
Send Byte,
Trflag
Trflag =Not Trflag
End of
File
Send EOF
End

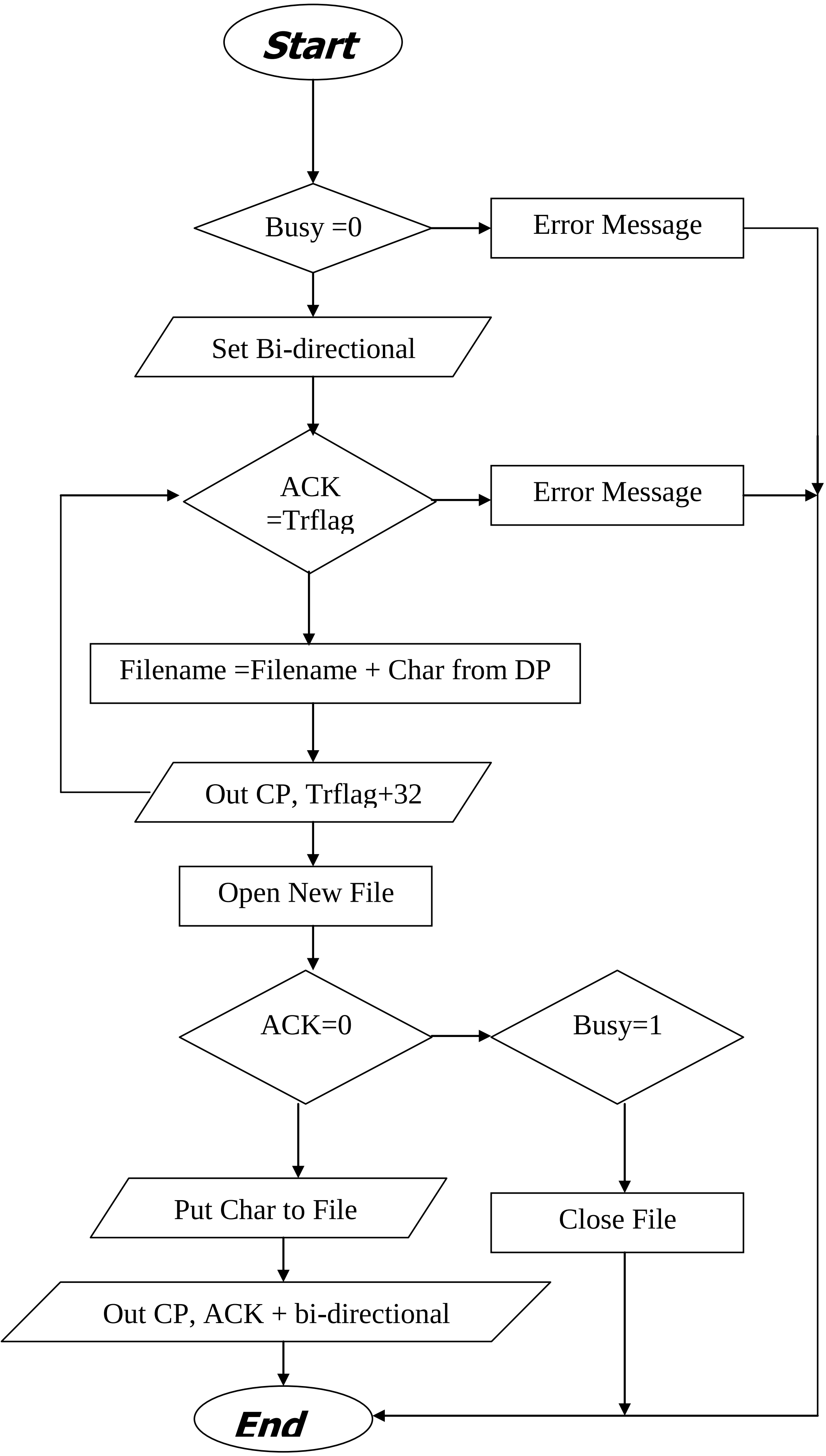
Send


Start
Trflag =1
Trflagc =1
Trflagc =0
Out DP, Byte
Out CP, Trflag
ACK
=Trflagc
Error
Send =-1
Send =0
End

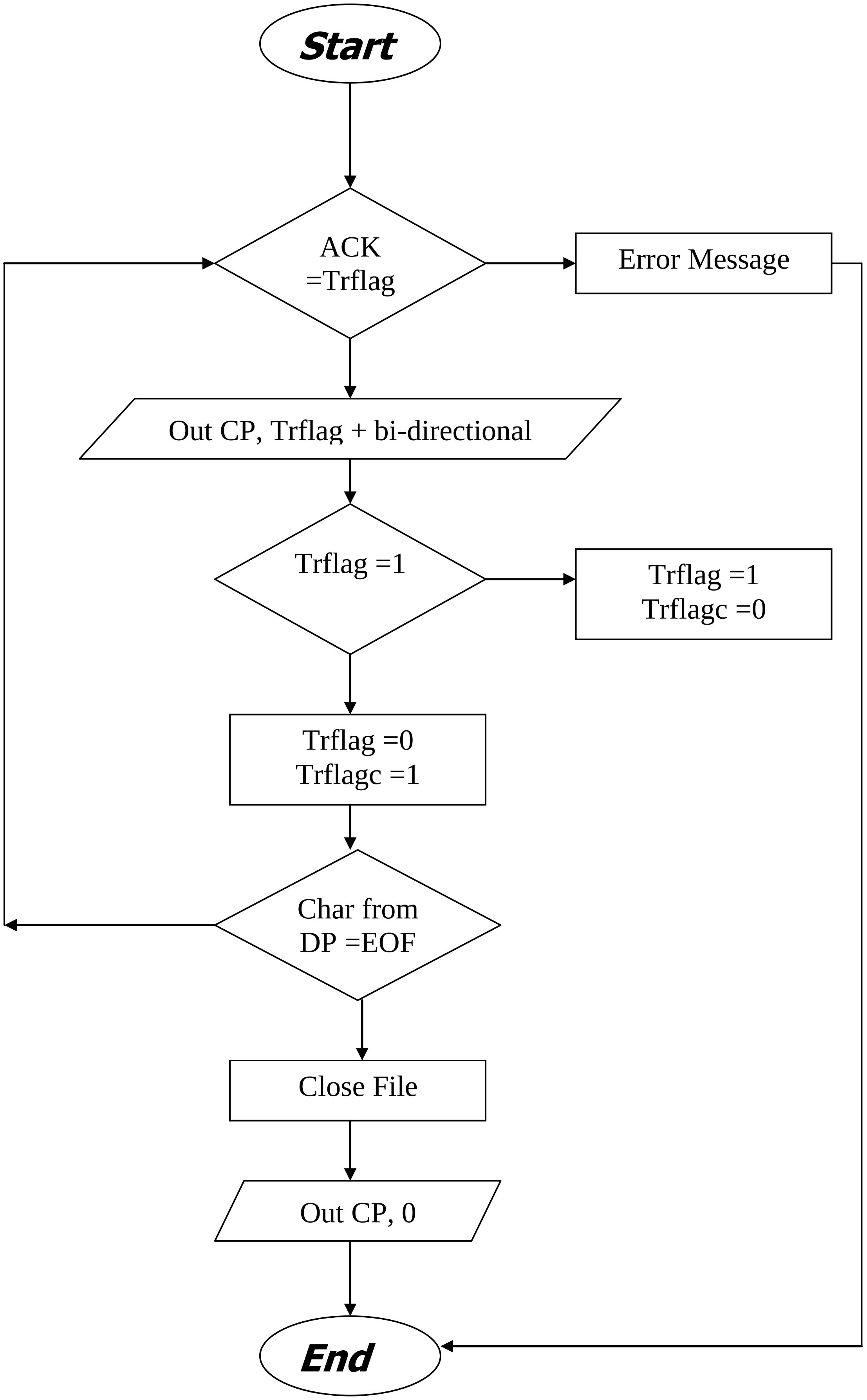
Check Status



Receive


Start
Busy =0
Error Message
Set Bi-directional
ACK =Trflag
Error Message
Filename =Filename + Char from DP
Out CP, Trflag+32
Open New File
ACK=0
Busy=1
Put Char to File
Close File
Out CP, ACK + bi-directional
End

Open Filename



Conclusions and Suggestions

From the importance details and results of transfer information and sharing between tow computers by EPP port have be seen.

Theoretical analysis and flowcharts were established for the EPP type of parallel port in order to increase the speed and the size of information when transferred which can be easily seen after the experiment done.

New hardware cable and special protocol as well as new software has been established, which was suitable and compatible with the computers.

An extension of this work could be making another protocol or using another language which may be more faster

References

[1] Criag Peakcock.*"Interfacing the Standard Parallel Port*".Copyright 1999-2001 19th August 2001

[2] Guy Hart-Davis." *Networking Complete*". Sybex, Inc., 2000

[3] Michael Tischer ." *PC Intern System Programming*". Data Becker 1995.

[4] Mike James ." *Microcontroller Cookbook".* Planta Tree, 1997

نقل البيانات بمقدار ثمانية بت عن طريق منفذ التوازي IC Epp

فراس شوكت حامد و قدامة عبد الجبار حامد و أحمد علي محمد صالح

قسم التقنيات الأليكتروني، المعهد الفني الموصل، الموص، العراق

المستخلص:

يتناول البحث تصميم وتنفيذ بروتوكول خاص لم يستخدم من قبل[١،٢] لعملية نقل البيانات بين حاسوبين (computer) باستخدام منفذ التوازي (Parallel Port) من نوع (EPP) على أساس نقل البيانات بمقدار 8 Bit.

تم إعداد برنامج يخضع تحت بروتوكول خاص بنا دون تقليد أحد البروتوكولات المطروحة في العالم[١،٢،٣] لنقل البيانات من والى الحاسوب مع اقل خطأ ممكن، حيث تم إعداده بلغة (Qbasic). يتألف هذا البرنامج من عدة دوال وهي دالة لواجهة تحديد وضعية الحاسوب هل في حالة إرسال أم استلام أم الخروج، ودالة للإرسال ودالة للاستلام ودالة لفتح ألملف المراد إرساله ودالة لفحص الـ (Bits) على منفذ التوازي ليمكننا من تحديد وقت الإرسال والاستلام وما هي الحالة التي هو عليها. يعمل هذا البرنامج بعد توفر (Data Cable) الذي تم تصنيعه تبعا للبرنامج الذي تم بناءه.