TRANSMISSION MODE HALF-DUPLEX AND FULL-DUPLEX IN MICROCONTROLLER ENVIRONMENT

Hanfoug Salah
Assistant professor, Department of telecommunication and electronics, University of Bejaia, Campus Targa Ouzemmour, Route de Targa Ouzemmour, Bejaia, 6000, Algeria
hanfougsalah@gmail.com
salah.hanfoug@univ-bejaia.dz

Abstract

This article discusses the basic principles of two communication protocol, the first on is the Serial peripheral interface (SPI) with transmission mode full-duplex the second on is I2C protocol with transmission mode half duplex, the SPI is one of the most widely used interfaces between microcontroller and peripheral ICs such as ADCs, DACs, sensors, SRAM, shift registers, and others. the I2C stands Inter-Integrated Circuit it is a bus interface connection protocol incorporated into devices for serial communication, the protocol of I2C is a half duplex supports multi-master bus system but more than one bus cannot be used simultaneously This article provides an explanation in detail of the SPI and I2C protocols, The SPI interface is a synchronous, full duplex-based interface. The data from the main is synchronized on the rising or falling clock edge. The SPI interface can be configured for either 3-wire or 4-wire. The popular 3-wire and 4-wire SPI interface is the focus of this work with the interference of processor and MAX7219 as application.

***Keywords*: Analog to digital converter, Full-Duplex, MAX7219, Protocol, sensors, serial, parallel, serial peripheral interface, Wireless communication.**

1. Introduction

With the rapid advancement of electronic and protocols of communication technologies [1], most of them, Smart homes infrastructure includes many devices and interconnected systems that can be used in a variety of applications. A networking paradigm known as information-centric networking (ICN) is one that is capable of maintaining packet delivery even in unreliable situations. As a result [2], the embedded systems use one or more processors or microcontrollers to perform specialized operations in a more complex system. These on-board controllers must communicate with other components, sensors, ADC and even system controllers. Although common, complex serial protocols and interfaces can be very difficult to program and troubleshoot, especially if the number of devices they communicate with is small. The Wireless Sensor Network (WSN) composed of tiny sensors distributed spatially, is such an infrastructure which is used to monitor and gather data about the physical situations of an environment or location. WSN collects the data using wireless sensors also called as nodes. Generally, the sensor node comprises of a microcontroller, analog-to-digital converter (ADC), transceiver, power source, and sensors [3].

In the microcontroller environment has three communication protocols [4], the first on is the asynchronous universal series port «USART». And the second on is the “PSP” parallel communication port and the last on is the "SSP" serial synchronous port with "SPI" and "I2C" protocols

The term UART corresponds to universal asynchronous receiver/ transmitter and defines a protocol, or set of rules, dedicated to the exchange of serial data between two devices. The UART is very simple and uses only two cables between the transmitter and the receiver to transmit and receive in both directions.

In “PSP” The data passes through the PSP0 to PSP7 lines, which physically use the same. The data flow is controlled by the RD, WR and CS lines it validates the slave by the CS (Chip Select) line, and the master it reads or writes using the RD (Read) and WR (Write) lines. Hence the name slave parallel port. Slave.

For the Module MSSP is a useful serial communication interface for exchanging information with other devices or microcontrollers. These devices can be a serial EEPROM, display drivers, ADC, DAC converters, The MSSP module can operate in one of two modes: Serial peripheral interface (SPI) and Inter-integrated circuit (I2C).

The electrical characteristics and communication protocol of I2C has become an industrial standard used by many manufacturers. It’s a synchronous and serial communication bus with a recognition protocol. The transfer

frequency is between 100 KHz and 400 KHz. The bus is controlled by a master who generates the communication clock. (It is possible to work in multi-master mode). All other circuits are slaves; they all receive the same clock from the master.

The SPI interface was developed by Motorola. It is a synchronous serial interface designed for short-distance communication between devices. Since then, it has become a de facto standard used by many semiconductor manufacturers, especially in microcontrollers and microprocessors. SPI is a common communication protocol used by many different devices. Such as, RFID card reader modules, , SD card reader modules and 2.4 GHz wireless transmitter/receivers.

In this paper, the SPI communication protocol in different mode of data transfer is proposed; each member of this transfer may issue and receive at the same time. This is made possible by a special organization of the transfer register and the SPI module. The figure below explains this idea.

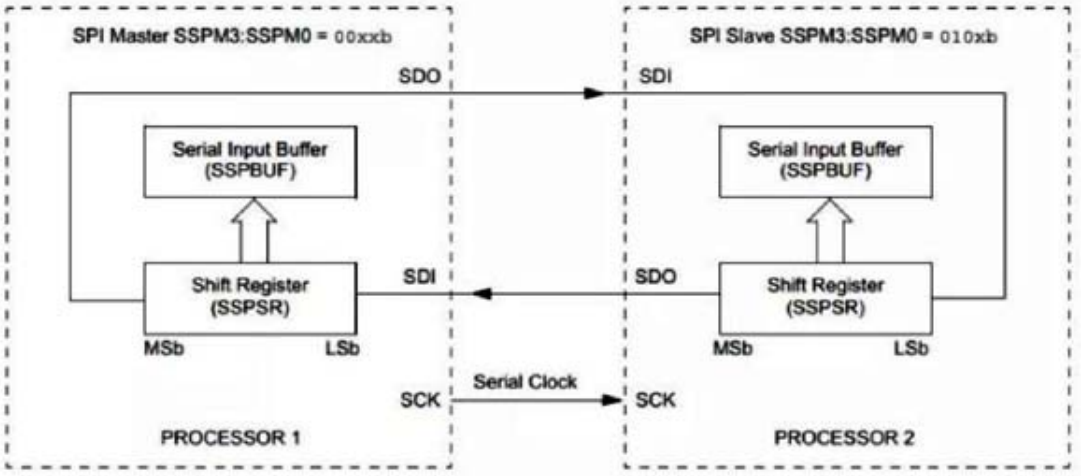

Fig 01 principal of communication full duplex [5]

The rest of this paper is organized as follows: In the following section, some related research is reviewed and discussed briefly. The SPI protocol is presented in Section 2. In Section 3, the transmission data in serial and parallel mode has been depicted in details. In section 4 evaluated the communication protocol by using application and simulation with MAX7219 and matrix display. Section 5 concludes the paper.

2. The SPI Protocol

The term SPI stands for Serial Peripheral Interface. It is a common communication protocol, used to exchange data, read/write, between multiple devices, only one of which must be configured as Master, usually a Microcontroller, and all others as Slavic. These can be SD cards, DAC, ADC, DS1306 Real Time Clocks, display controllers (MAX2719 serial display driver), 25LC256 Serial EEPROM, smart sensors, shift registers, and so on [6]. Such as the sensor nodes are also equipped with sensors, which are hardware devices capable of measuring the change in the physical conditions of surroundings like temperature, pressure, etc. ADC is deployed to convert analogue values to the digital signals [2].

Any exchange of information should go through the Master between the Master and the Slavs the transfer of information is direct, but if two Slavs plan to communicate with each other, they will have to use the Master as an intermediary. SPI serial communication is SYNCHRONOUS and FULL DUPLEX (full duplex). Synchronous because it works with a clock signal shared between the members of this configuration, and this clock is provided by the Master. In addition to its higher data transfer speed, SPI also uses a very simple data transfer protocol compared to other serial data transfer methods. The Master uses the same register (8 bits) to send/receive data from the Slavs, and each uses its own register (8 bits) to send/receive data from/to the Master. The complete transfer of a character is done in only 8 clock cycles, which would consume 0.8 us if we hold the 10 MHz clock.

The reason for the popularity of SPI lies in its many advantages. First, it is a simple hardware addressing interface that offers total flexibility on the number of bits transferred. it uses a master-single master slave and can handle multiple slave devices using full duplex communications with clokc frequencies up to 50 MHz. It does not use standard protocol and only transfers data packets, which makes it ideal for transferring long data streams. The SPI uses a maximum of four signal lines (Figure 2).

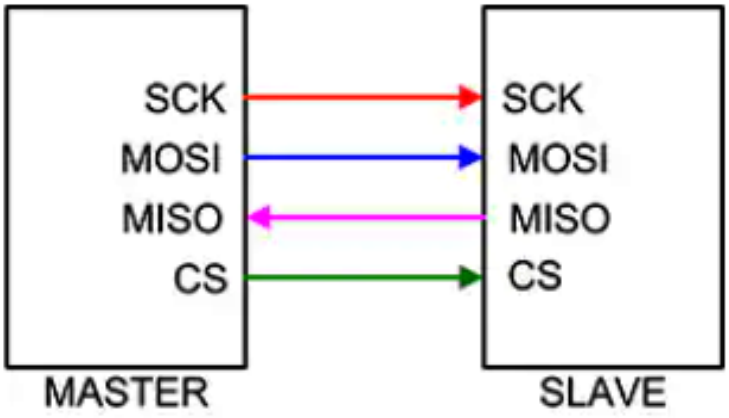

Fig 02: The full duplex SPI connection uses two data lines (MOSI, MISO), a clock line (SCK) and a chip selection line (CS).

The master device, usually a processor or controller, provides and controls clock lines (SCK) and chip selection (CS). Full multiplex operation is managed by MOSI (output master, slave input) and MISO (master input, slave output) data lines. In a simple single master and single slave configuration, the chip selection line can be eliminated and the CS input to the forced slave in the active logical state. If the slave device can only send data (semi-duplex communication), then the MOSI line can also be deleted to further reduce the number of signals. The data is transferred by the clock signal, so that the data transfer is assimilated to an offset register with a staggered bit for each clock.

There are some advantages to using SPI, and if given the choice between different communication protocols, such as, No start and stop bits, so the data can be streamed continuously without interruption, No complicated slave addressing system like I2C, Separate MISO and MOSI lines, so data can be sent and received at the same time and Higher data transfer rate than I2C (almost twice as fast). and for disadvantages in SPI protocol Uses four wires (I2C and UARTs use two), No form of error checking like the parity bit in UART and no acknowledgement that the data has been successfully received.

3. The I2C Protocol

The electrical characteristics and communication protocol were I2C has become an industrial standard used by many manufacturers. It's a synchronous and serial communication bus with a recognition protocol. The transfer frequency is between 100 KHz and 400 KHz. The bus is controlled by a master (MASTER) who generates the communication clock. (It is possible to work in multi-master mode). All other circuits are slaves (SALVES), they all receive the same clock from the master.

In the I2C the transmission begins with a start-condition (s), followed by the address, code on 7 or 10 bits, completed by the R/W bit, which specifies by signal data to written or read by the master. Each byte sent is always accompanied by an acknowledgement of receipt from the receiver. On byte therefore requires 8+1=9 clock pulses on the SCL pin. Logically, it is always the master that sends the address, whether it is the receiver or the sender for the rest of the message.

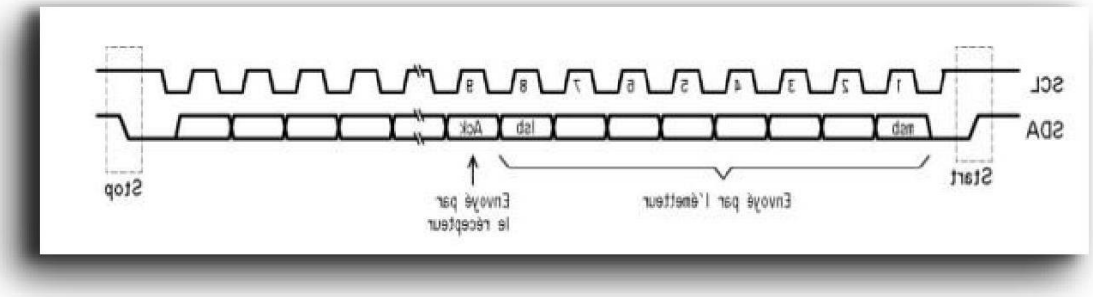

Fig 03 frame structure (communication I2C) [5]

4. Full-Duplex Mode

In cellular networks, there are two types of wireless full-duplex communications: symmetric and asymmetric. In symmetric full-duplex communication [7], the transmission occurs between two nodes (usually the base station (BS) and one user equipment. This kind of SCFD is sometimes called bidirectional. On the other hand, the asymmetric half-duplex includes three nodes [8,9]. Full-Duplex mode most closely matches the operation of the SPI incorporated into the MSSP module of processors devices, allowing for simultaneous exchange of data between the master and slave devices. To configure the SPI module as a master in Full-Duplex mode, the RXR and TXR registers of the SPIxCON2 register must both be set high. Depending on the status of the BMODE bit, the transfer counter may need to be loaded to initiate the SPI module to begin transferring data [10].

5. Transmission data in serial and parallel mode

The SPI Bus can have several Slavs each with its own SS (Chip Select), and the Master should have enough lines to independently select each of them. The number of these lines depends on the number of Slavs. Such as two configurations are possible

5.1 Configuration in loop (DAISY CHAIN)

In the loop configuration (DAISY CHAIN), all members are connected as a circular structure, and data must flow from member to member to their final destination. Each device is connected between 2 other devices so that the whole gives rise to a logical circle. A single SS line is then required to establish communication in this configuration as shown in the diagram below

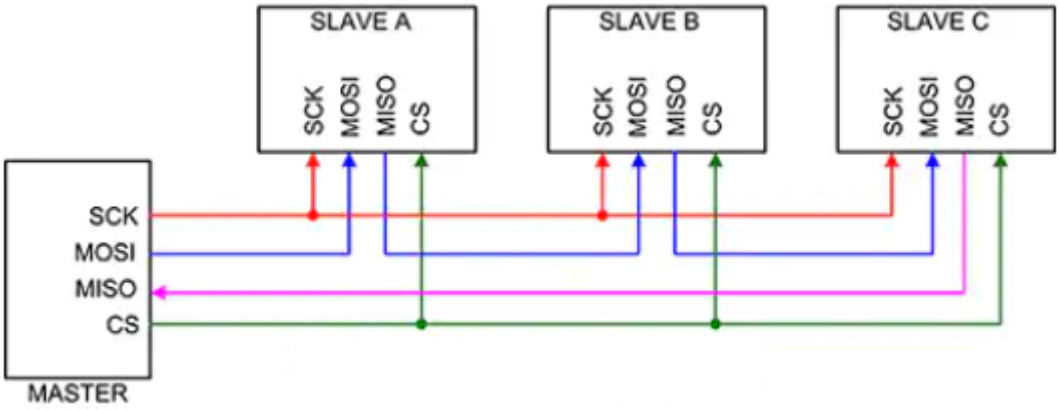

Fig 04: Configuration en mode DAISY CHAIN.

This type of structure is usually used in output-only situations, such as checking LEDs (LEDs) for which there is no feedback signal to the Master. In these cases, the SDI line (MISO) of the Master can be left disconnected. However, if the data must be returned to the Master, it can be done by closing the loop (daisy chain). Note that if this is done, the data returned by Slave 1 must propagate along all other Slavs before returning to the master. So you have to make sure that you send enough intake orders to get the data you need.

5.2 Configuration in bus

In this configuration the devices of this configuration are arranged in Bus, and the number of selection lines that the Master must have to manage the communication, is a function of the number of Slave played. All Slavs are listening, but only the Slav who sees his SS (CS) in the low state is concerned by the transfer. Since the Slavic SDO/SDI pins are connected together, they must be three-state logical pins. Slaves devices whose pins do not support all three states can be used by adding buffers to three states controlled by the Chip Select signal. as shown in the diagram below

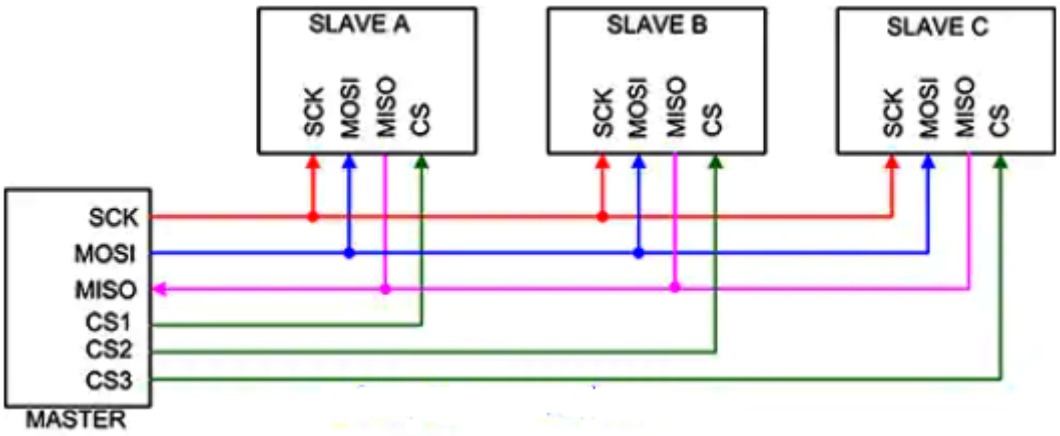

Fig 05: direct configuration

6. Interferences MAX7219 in serial and parallel mode

6.1 THE SCHEMA SYNOPTIC OF DISPLAY SYSTEM

The MAX7219 and MAX7221 integrated circuits are serial interface display controllers. These components are able to drive up to 8 7-segment displays with decimal point or 64 LEDs while ensuring multiplexing between the different digits as well as decoding the values to be displayed. The block of schema synoptic as represented in figure (05) is gives an overview of the different parts in system display, which is divided into four main parts, the power supply, processing control unit (circuit integrated MAX7219, processor), controls button and display unit.

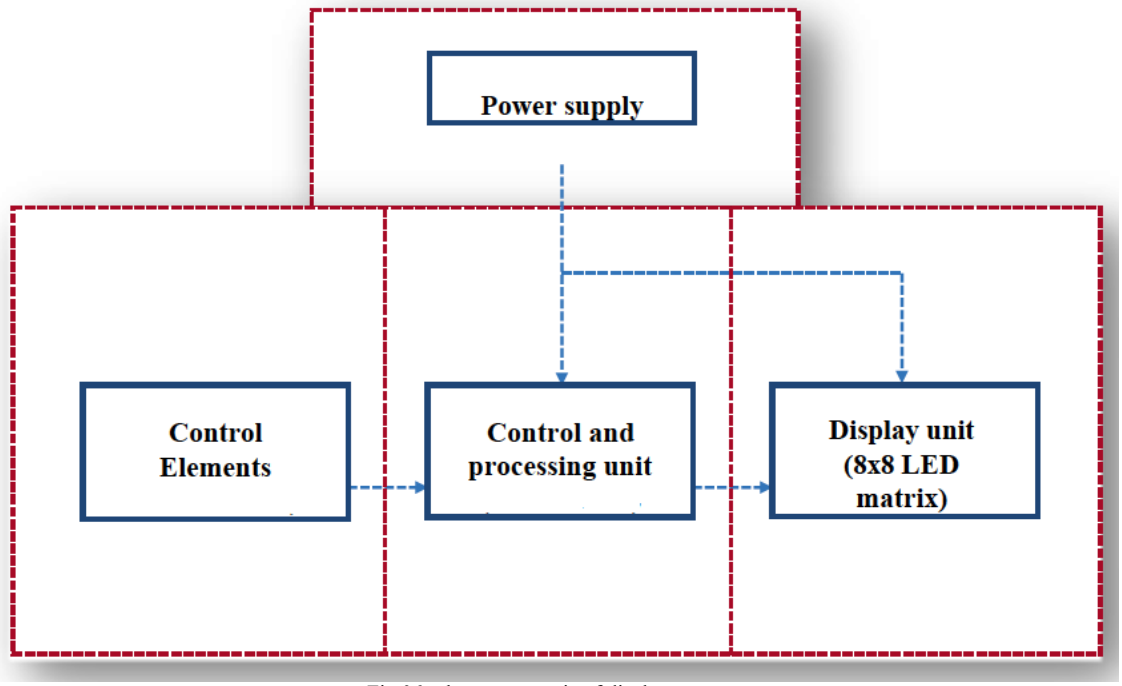

Fig 06 schemas synoptic of display system

6.2 THE ELECTRONICS SCHEMATIC OF DISPLAY SYSTEM

The SPI protocol characterised by full duplex communication, independent of the number of bits to be transmitted, no collision possible, slaves use the master clock, much faster than I2C protocol in standard mode and multi-master configuration is possible. When configuring the SPI module on a processor device, the first step should be to ensure that all devices have been connected correctly in hardware on the SPI bus. Once the master device has been connected to all slave devices in hardware, the next step is to assign necessary pins for SPI communication in software. To vary the animation of the texts displayed on the matrix led with tow configuration. We designed two different montages in serial and parallel mode as showing in figure 06 and figure 07

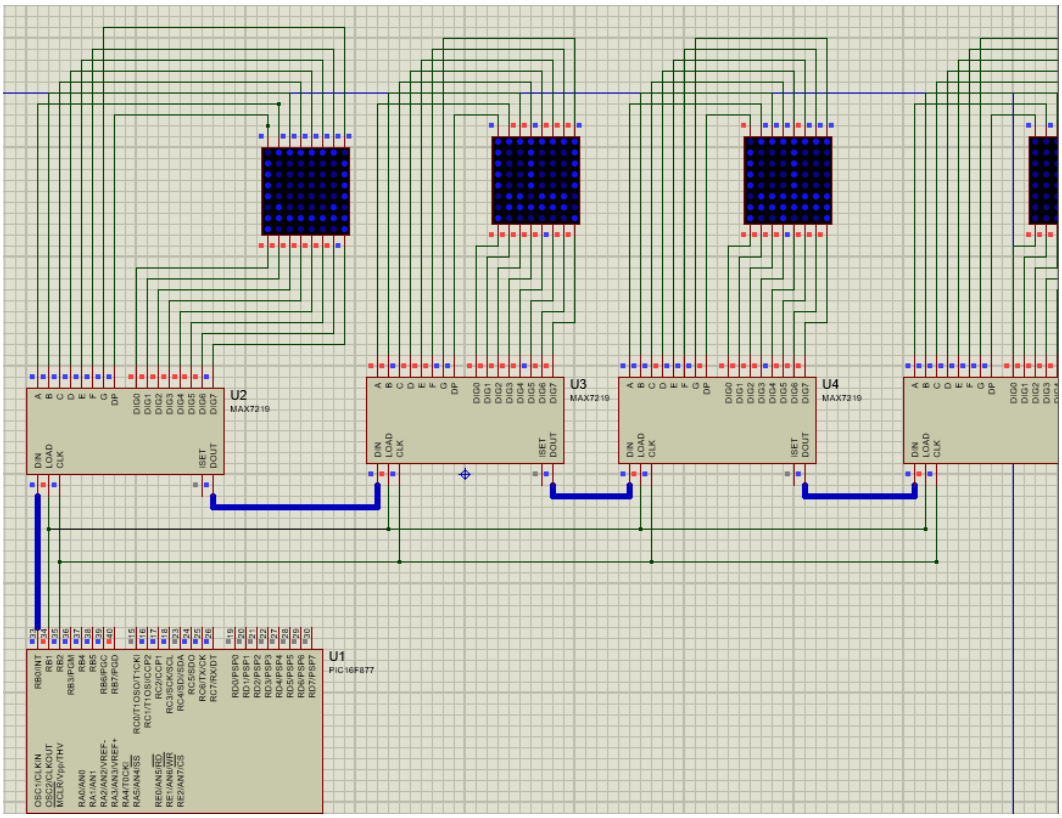

Figure 07 Serial configuration of SPI protocol

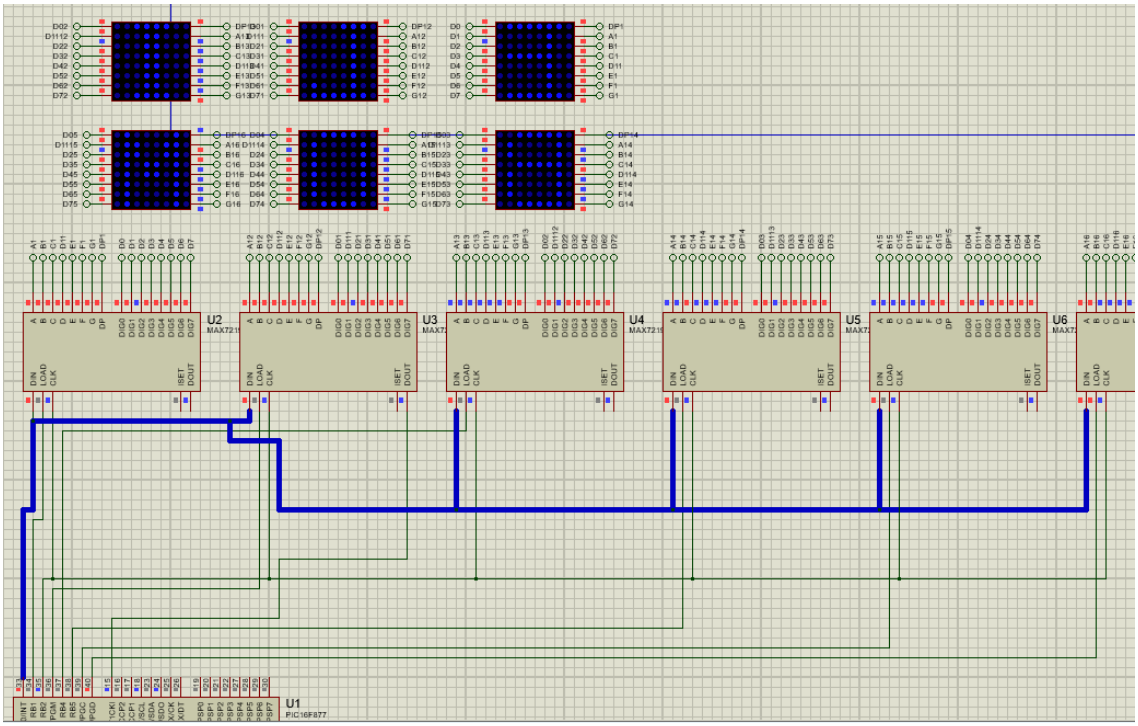

Fig 08 Parallel configuration of SPI protocol

6.3 THE FLOWCHART OF DISPLAY SYSTEM

The figure below interdicted the flowchart of main program with subroutines, disrupted general representation of display application. Such as must configuration the pin as output or input and port initialization is MOSI, MISO, CS and CLK, after initialization the matrix used by MAX7219 for the display of alphabet and symbols. For the subroutine. The first on for the shift register and the second on for display in the matrix.

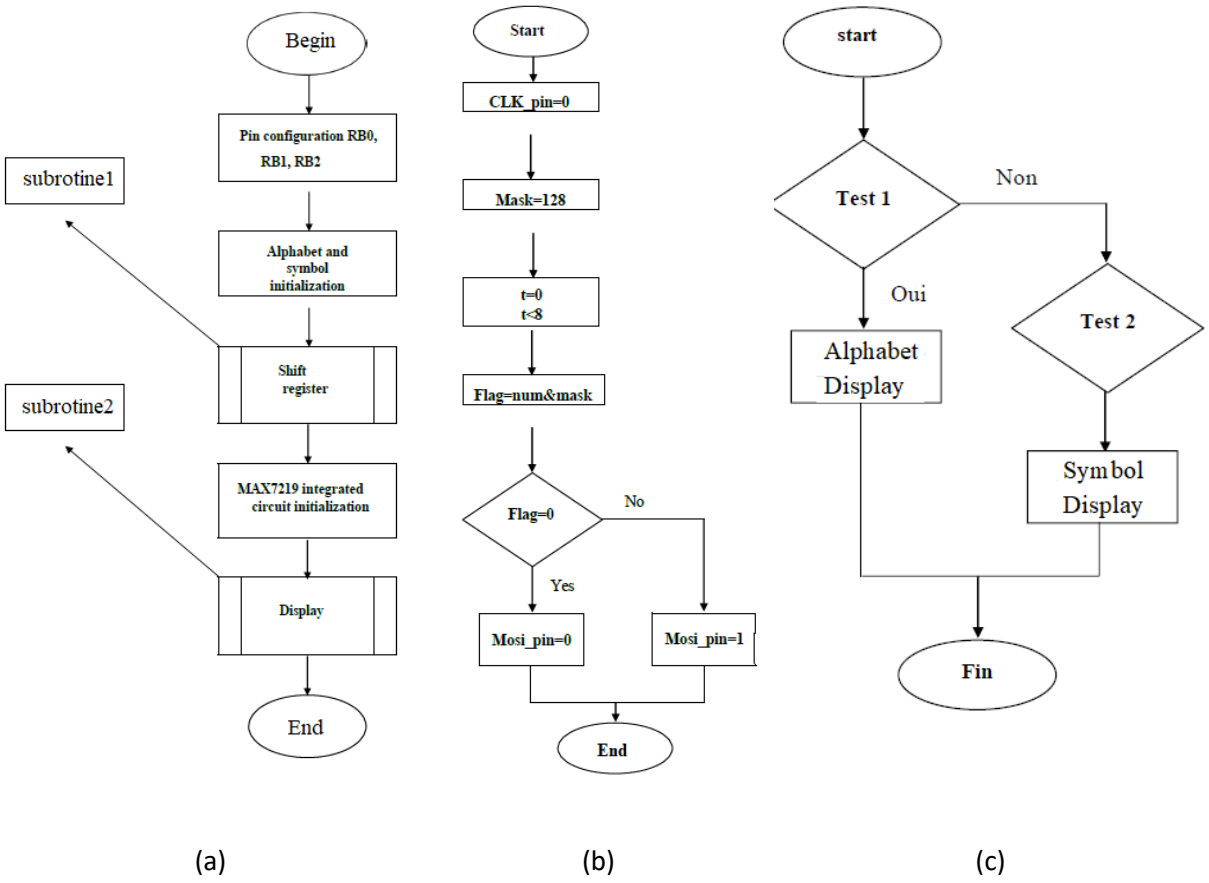

Fig 09 (a)Main program (b) subroutine 1, (c) subroutine 2

In the SPI protocol any exchange of information should pass through the master and slave, the transmission of data is direct, but if two slaves intend to communication with each other, they will have to use the master as an intermediary, spi serial communication is synchronous and full duplex it's also uses a very simple data transfer protocol compared to other serial data transfer methods, the master uses the same register to send and receive data from the slaves, and each uses its register to send and receive data to and from the slaves, The step-by-step algorithms for the spi protocol in to mode coordinator and the data transfer procedure for to mode serial and parallel are shown in Figure 10 and Figure 11, respectively.

```
void send(unsigned short ss, int dig1,int dig2,int dig3,int dig4)
{switch(ss)
 {case 0:
 St1 = (dig1 - 65) * 6;
for(x=1;x<9;x++)
{Cs_pin1 =0;
SPI1_write_Byte(x);
SPI1_write_Byte(Alphabet[st1]);
delay_ms(40); Cs_pin1 =1; st1++;
} break;
} switch(ss);
 {case 2:
 St1 = (dig1 - 65) * 6;
 St2 = (dig2 - 65) * 6;
for(x=1;x<9;x++)
{  Cs_pin1 =0;
SPI1_write_Byte(x);
SPI1_write_Byte(Alphabet[st1]);
delay_ms(40);
Cs_pin1 =1;
Cs_pin1 =0;
SPI1_write_Byte(x);
SPI1_write_Byte(Alphabet[st2]);
SPI1_write_Byte(0x00);
SPI1_write_Byte(0x00);
delay_ms(40);
Cs_pin1 =1;
st2++; st1++;
} break;
 } switch(ss)
 {case 3:
 St1 = (dig1 - 65) * 6;
 St2 = (dig2 - 65) * 6;
 St3 = (dig3 - 65) * 6;
for(x=1;x<9;x++)
{  Cs_pin1 =0;
SPI1_write_Byte(x);
SPI1_write_Byte(Alphabet[st1]);
delay_ms(40);
Cs_pin1 =1;
Cs_pin1 =0;
SPI1_write_Byte(x);
SPI1_write_Byte(Alphabet[st2]);
SPI1_write_Byte(0x00);
SPI1_write_Byte(0x00);
delay_ms(40);
Cs_pin1 =1;
Cs_pin1 =0;
SPI1_write_Byte(x);
SPI1_write_Byte(Alphabet[st3]);
SPI1_write_Byte(0x00);
SPI1_write_Byte(0x00);
SPI1_write_Byte(0x00);
SPI1_write_Byte(0x00);
delay_ms(40);
Cs_pin1 =1;
st2++;st3++; st1++;} break;
```

Fig 10 algorithm for transmission serial mode for SPI protocol

```
VOID SEND(UNSIGNED SHORT SS, UNSIGNED SHORT _DATA)
{SWITCH(SS)
 {CASE 1:
TRISB.F3 = 1;  TRISB.F1 = 0; TRISB.F4 = 1; TRISB.F5 = 1;  TRISB.F6 = 1; TRISB.F7= 1;
WRITE_CHAR(_DATA);
 TRISB.F1 = 1;
 BREAK;
 CASE 2:
 TRISB.F3 = 0;  TRISB.F1 = 1; TRISB.F4 = 1; TRISB.F5 = 1;  TRISB.F6 = 1; TRISB.F7= 1;
 WRITE_CHAR(_DATA);
 TRISB.F3 = 1;
 BREAK;
CASE 3:
TRISB.F1 = 1; TRISB.F4 = 0; TRISB.F5 = 1; TRISB.F3 = 1; TRISB.F6 = 1; TRISB.F7= 1;
WRITE_CHAR(_DATA);
TRISB.F4 = 1;
 BREAK;
CASE 4:
 TRISB.F1 = 1; TRISB.F4 = 1; TRISB.F5 = 0; TRISB.F3 = 1; TRISB.F6 = 1; TRISB.F7= 1;
  WRITE_CHAR(_DATA);
TRISB.F5 = 1;
TRISB.F5 = 1;
BREAK;
CASE 5:
TRISB.F1 = 1; TRISB.F4 = 1; TRISB.F5 = 1; TRISB.F3 = 1; TRISB.F6 = 0; TRISB.F7= 1;
WRITE_CHAR(_DATA);
TRISB.F6 = 1;
 BREAK;
 CASE 6:
TRISB.F1 = 1; TRISB.F4 = 1; TRISB.F5 = 1; TRISB.F3 = 1; TRISB.F6 = 1; TRISB.F7= 0;
  WRITE_CHAR(_DATA); DELAY_MS(5000);
TRISB.F7= 1;
 BREAK;
  CASE 22:
 TRISB.F1 = 0; TRISB.F3 = 0;TRISB.F4 = 0; TRISB.F5 = 0;  TRISB.F6 = 0; TRISB.F7= 0;
   CLEAR_MATRIX();DELAY_MS(50);
 TRISB.F1 = 1; TRISB.F3 = 1;TRISB.F4 = 1; TRISB.F5 = 1;  TRISB.F6 = 0; TRISB.F7= 1;
 BREAK;
CASE 7:
 TRISB.F1 = 0; TRISB.F4 = 1; TRISB.F5 = 1; TRISB.F3 = 1; TRISB.F6 = 1; TRISB.F7= 1;
  WRITE_CHAR(_DATA);
  TRISB.F1 = 1;
 BREAK;
 CASE 8:
 TRISB.F1 = 1; TRISB.F3 = 0;TRISB.F4 = 1; TRISB.F5 = 1;  TRISB.F6 = 1; TRISB.F7= 1;
  WRITE_CHAR(_DATA);
  TRISB.F3 = 1;
 BREAK;
 CASE 9:
 TRISB.F1 = 1;  TRISB.F3 = 1;TRISB.F4 = 0; TRISB.F5 = 1; TRISB.F6 = 1; TRISB.F7= 1;
  WRITE_CHAR(_DATA);
  TRISB.F4 = 1;
 BREAK;
 CASE 10:
 TRISB.F1 = 1; TRISB.F4 = 1; TRISB.F5 = 0; TRISB.F3 = 1; TRISB.F6 = 1; TRISB.F7= 1;
  WRITE_CHAR(_DATA);
  TRISB.F5 = 1;
 BREAK;
 CASE 11:
:
```

Fig 10 algorithm for transmission parallel mode for spi protocol

7. Conclusion

The transmission of information from the processor to the display devices using the MAX7219 integrated circuit via the SPI communication protocol is established, The SPI protocol solves the need for a simple or complex configuration, inexpensive and low-overload simple interface in applications where the source can be assimilated to a data stream, as opposed to reading and writing data to address locations. This is why it is ideal for managing device-to-device communications between microcontrollers and sensors, digital signal processing devices, and other processors

References

[1] Sabin Bhandari & Sangman Moh, (2019)., A mac protocol with dynamic allocation of time slots based on traffic priority in wireless body area networks, International Journal of Computer Networks & Communications, Vol.11, PP 25-41.
[2] Yousra Abd Mohammed and Shayma Wail Nourildean, (2023), An Intelligent Encrypt/Decrypt Algorithm in IoT Regarding Home Data Privacy and Security , International Journal of Computing and Digital Systems Vol.13, pp1352-1358
[3] Anuj Kumar Singh1 and B.D.K.Patro, (2019), "A NOVEL SECURITY PROTOCOL FOR WIRELESS SENSOR NETWORKS BASED ON ELLIPTIC CURVE SIGNCRYPTION", International Journal of Computer Networks & Communications (IJCNC) Vol.11, pp93-112.
[4] Christian Tavernier (2005), Les microcontrôleurs PIC : recueil d'application, 3éme édition, Dunod.
[5] Han-Way Huang (2004), PIC Microcontroller An introduction to software and hardware interfacing, 1st edition, THOMSON Delmar learning
[6] Piyu Dhaker, (2018), Introduction to SPI Interface, Analog device.
[7] Ali Y. Al-Zahrani, (2018)" Modelling and QoS-Achieving Solution in full-duplex Cellular Systems", International Journal of Computer Networks & Communications (IJCNC) Vol.10, pp117-135
[8] A. Sabharwal, P. Schniter, D. Guo, D. W. Bliss, S. Rangarajan, and R. Wich¬ man (2014), In-band full-duplex wireless: Challenges and opportunities, IEEE Journal on Selected Areas in Communications, vol. 32, pp 1637-1652.
[9] B. P. Day, A. R. Margetts, D. W. Bliss, and P. Schniter (2012), "Full-duplex mimo relaying: achievable rates under limited dynamic range," IEEE J. Sel. Areas Commun., vol. 30, pp. 1541– 1553.
[10] Jason Layton, (2018), Using the SPI Module on 8-Bit PIC Microcontrollers, Microchip Technology Inc.