Proceedings of 4th RIT Post Graduates Conference (RIT PG Con-18)
NOVATEUR PUBLICATIONS
JournalNX- A Multidisciplinary Peer Reviewed Journal (ISSN No:2581-4230)
April, 13th, 2018

DESIGN AND IMPLEMENTATION OF I2C PROTOCOL

Ms. Kamble Anuja Ashok

Department of Electronics and Telecommunication Engineering
Rajarambapu Institute of Technology,
Islampur, India.

anuja7292@gmail.com

Dr. S. A. Pardeshi

Department of Electronics and Telecommunication Engineering
Rajarambapu Institute of Technology,
Islampur, India
sanjay.pardeshi@ritindia.edu

***Abstract*—The serial communication plays an important role in various applications. By using the communication protocol, we increase the performance of a system. Popular serial interfacing protocol includes: USB, I2C, SPI, CAN and UART for communication between integrated circuits for low/medium data transfer speed with on board peripherals. Our system allows the secure exchange of large amount of data. A protocol handles fragmentation, encryption, error handling, session handling and authentication. Our target is an implement for industrial applications to write a configuration or firmware or to read parameter from the devices. In this paper design and implementation of I2C protocol is presented. In that single master and multiple slave are used to design I2C bus controller. The hardware implementation of the I2C protocol is done using Arduino Nano.**

Keywords—I2c bus controller, multi slave, serial data (SDA), serial clock (SCA), multimaster

I. INTRODUCTION

I2c bus invented in 1982 by Philips Semiconductor. Philips developed simple bi-directional 2-Wire bus which is called Inter-IC control or also I2C bus controller. In that only two lines are required that is SDA (Serial Data Line) and SCL (Serial clock line). SDA and SCL line carry information between devices which is connected on the bus. It is widely accepted as a serial communication protocol. I2C bus is used for connecting low speed devices like EEPROM's ADC, DAC and Microcontroller, etc. I2c have different speed mode, for Standard mode 100kbits/s, for Fast mode 400kbit/s and high-speed mode 3.4Mbit/s.

This paper describes a platform of I2C communication protocol implementation using Arduino Nano board. In this system data exchange of devices takes place. In this project, single master and multiple slaves are implemented. There are two conditions in I2C protocol that is STARI condition and STOP condition which are always generated by the master. In START condition high to low transition on the SDA line while SCL is high and in STOP condition a low to high transition on the SDA line while SCL is high.

II. LITERATURE REVIEW

Anagha A, M.mathurankani et al. [1] have presented dual master – dual slave I2C bus controller. It is implemented on Spartan 3A FPGA. In that paper EEPROM 24C07 and 2408 is used as the slave devices. It is developed based on the FSM (Finite State Machine) technique using VHDL language. Initially they developed for single master bus controller later it is extended for dual master bus controller. EEPROM24CXX series of 256 byte is used as a slave for simulation. In this writing to the memory location of EEPROM and reading from the EEPROM takes place. In that device address for EEPROM 24C07 is 1010000 and EEPROM 24C08 is 1010001. If two master simultaneously tries to access the slave, then master 1 will won the bus because the device having lowest binary value. So, in this project design of the multimaster I2C protocol is developed using Spartan 3A FPGA board.

Jacob Maxa, Thilo Kranchenfels et al. [2] have presented wireless packet – based serial data transmission using an EEPROM with I2C and NFC interface. Here secure exchange of large amount of data takes place. NFC is a subset of a RFID, which is used for the point-to-point communication in small data exchange. In NFC technology magnetic field is used for transmitting power and data from the transmitter to the receiver. The aim of this project is data exchange between a mobile device and an industrial device. This protocol handles data flow, fragmentation and error detection. It will secure every single packet with a password authentication and CRC checksum for error detection.

Aruna Kommu, Raghavendra Rao Kanchi et al. [3] have presented implementation of communication protocol such as USB, I2C, SPI/SSP, CAN and URAT. ARM Microcontroller PLC2148 hardware platform is used for the implementation. This system allows the data exchange between two devices. This paper helps the student to implementation of serial protocol using ARM7TDMI. Here, LPC2148 is used as a master and AT24c04 is used as a slave.

Yassine Bouterra, Alaa chabir et al. [4] have presented distributed architecture control of an industrial robot. This design is based on the real time – multitasking control low system. The architecture is distributed in five microcontroller with master salve topology. To network management I2C protocol is developed. From LABVIEW instruction are introduced and transmitted to the master via USB connection. The master card is designed in embedded microcontroller PIC 18F2550. A control law for trajectory tracking is implemented in each slave controller. So, low – cost control system with master slave network has been developed. An embedded PIC18F25550 board is used as a master. A distributed PID controller are implemented using a microcontroller network. PIC 18F458 is used as a slave.

III. OVERVIEW OF I2C PROTOCOL

I2C stands for the Inter Integrated circuit. It is a multimaster bus controller which is developed by the Philips. The key features of I2C are:

- It requires only two signal lines i.e. SDA and SCL.
- A repeated START condition and STOP condition.
- It is collision detection and arbitration to prevent data corruption.
- It is 8-bit data.
- Flexible data transmission rate.
- ICs can be added to or removed from a system without affecting any other circuit on bus.
- It is bi-directional protocol.

The block diagram of I2C bus as shown in Fig. 1. It establishes a master and slave relationship. Master it is a host node which is control over the devices which is connected on bus. Master want to request data from the slave devices. Slave is a device which respond to the master. There are only two wires, SDA and SCL are used for carrying information between connected devices. The device connected on the bus have unique address. All slave have unique address but it is optional for the master node. It is a 7-bit address which is followed by the single bit to represent the write and read operation. If logic 0 then write operation carried out and if logic 1 read operation is carried out.

The two types of addressing mode are present in the I2C bus, a 7-bit addressing mode and 10-bit addressing mode. In 7-bit addressing mode 128 devices can be connected to the bus and in 10-bit addressing mode 1024 devices can be connected to the bus.

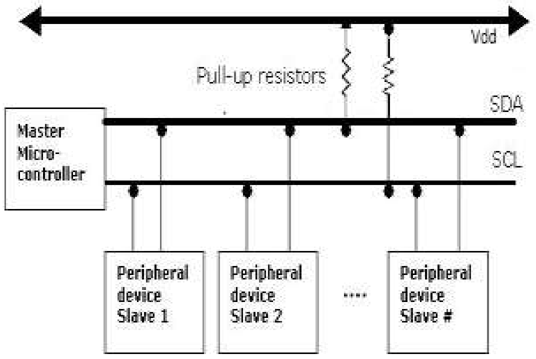

Vdd
Pull-up resistors
SDA
SCL
Master Micro-controller
Peripheral device Slave 1
Peripheral device Slave 2
....
Peripheral device Slave #

Fig.1 Block diagram of basic I2C bus

Data transmission in I2C:

In IDEL condition, SCL and SDA are at logic 1. If the bus is IDLE, then master can initiate a START signal. In that HIGH to LOW transition on the SDA line while SCL is HIGH state as shown in Fig.2.

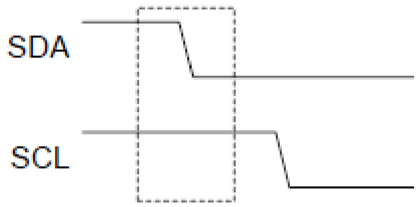


Fig.2 START condition

The bus is considered to be busy after START condition. The bus is considered to be free again certain time after STOP condition. In STOP condition LOW to HIGH transition on the SDA line while SCL is HIGH state as shown in Fig.3.

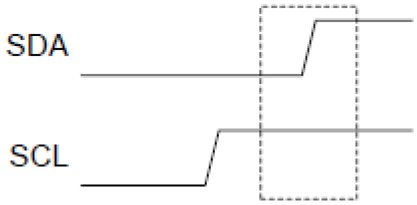


Fig.3 STOP condition

The frame format of I2C bus data transmission protocol as shown in Fig4.

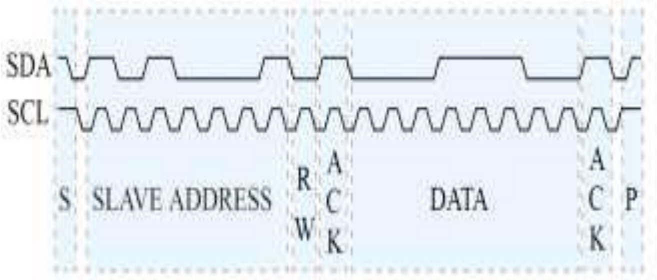


Fig.4 frame format of I2C bus data transmission

IV. DESIGN AND IMPLEMENTATION OF I2C PROTOCOL

In this work, I2C bus controller was developed by using Arduino Nano. The ATmega328 is used as it is simple, low-powered and low cost. There is one master which is host node and multiple slave devices are connected to the bus. On slave side analog sensors are used such as PT100 sensor and ACS712 current sensor is used. PT100 is used for the measure the temperature by changing its resistance depending on the surrounding temperature it is a slow response but provided accurate value. RTD resistance consist of length of fine wire wrapped around a ceramic or glass core. Here we use PT100 with range of 0^0C to 100^0C.it has a resistance 100ohms at 0^0C and 138.4 ohms at 100^0C. ACS712 current sensor is used to measure AC/DC current up to 20A. it has low-noise analog signal path, 80kHz bandwidth. It is operate from 5V and output voltage proportional to AC or DC current.

In I2C for master device address is optional whereas for slave devices there is unique address. For PT100, slave device ID is given as 1 and for ACS712 current sensor slave device ID is 2. For connecting slave devices with master node only two wires are requires, namely SDA and SCL.

The master device has control over the slave devices. Master want to request the data from the all connected slave devices and salve transmit the corresponding data to the master device. The design steps of implementation of this protocol as follows:

A. Discover node
B. Get data

A. Discover node:

In that step how many number of slave devices are connected to the master is find out. Here master scan the bus for how many devices are present on the bus. On I2C bus up to 127 devices are to be connected. For that master send the discover node command to the connected slave devices which are present on the bus. From that we find the how many devices are connected to the master and respective address of slave devices. Slave receive the command transmit the corresponding slave ID.

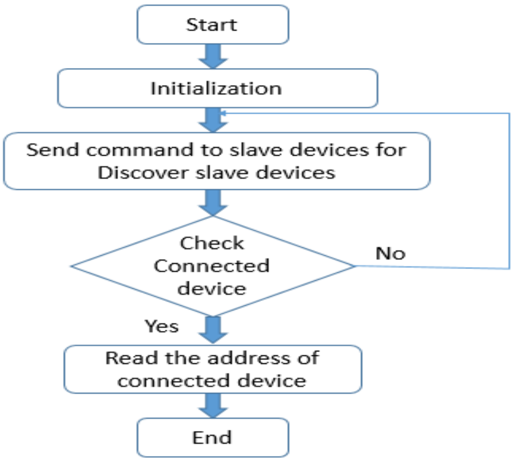

Start
Initialization
Send command to slave devices for Discover slave devices
Check Connected device
No
Yes
Read the address of connected device
End

Fig.5 flow char for Discover node

B. Get data:

In that master want to read the analog value of connected sensor. Which is connected to the analog pin of the Arduino Nano board on slave side. Master send the get data command to the connected devices. If command receive by the slave device then slave send the sensor value to master device on I2C bus.

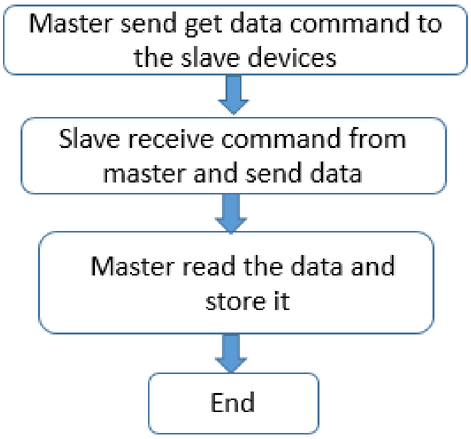

Master send get data command to the slave devices
Slave receive command from master and send data
Master read the data and store it
End

Fig.6 flow chart of get data

From above step, implementation of I2C protocol is carried out. For implementation of I2C network on Arduino board A4 is used for the serial data (SDA) and A5 is used for the serial clock (SCL). General connection for I2C network are shown in following Fig.7.master device start I2C communication by sending START bit and the slave device send ACK bit to the master. The data is transmitted between START and STOP condition. Arduino c language used for the programming algorithm.

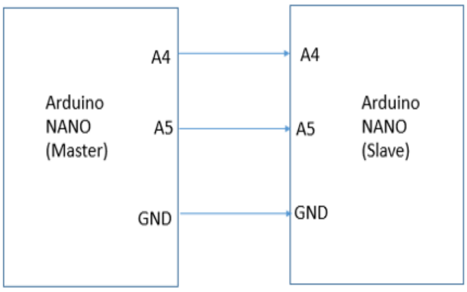


Fig.7 Establishment of I2C network using Arduino

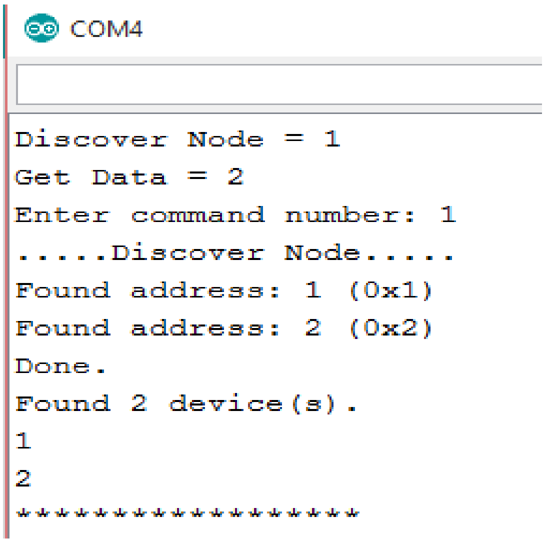
V. RESULT

The result of I2C protocol are as follows. In that master send command to the slave device, then slave will act on particular command.

A. Discover node output

Here. It will found that how many devices are connected to the master device and also know the IP address of the connected slave devices. In that project two salve devices are connected to the master.

Fig. 8 shows the output of discover node. Here we found the address of connected devices. PT100 sensor connected to the address 1 and ACS712 current sensor connected to the address 2. Fig. 8 shows that connected devices addresses in decimal and hex. Also shows the total number of device found.

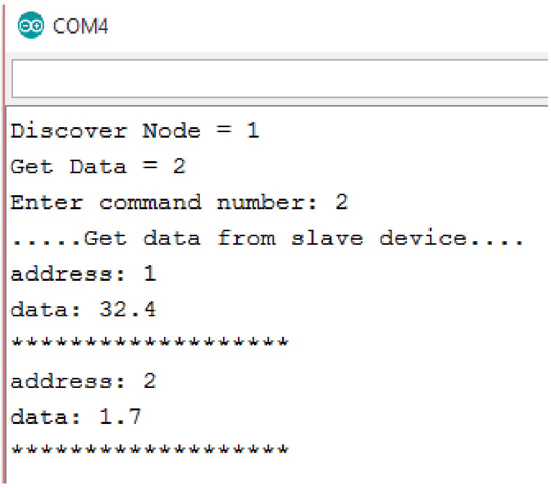


```
Discover Node = 1
Get Data = 2
Enter command number: 1
.....Discover Node.....
Found address: 1 (0x1)
Found address: 2 (0x2)
Done.
Found 2 device(s).
1
2
******************
```

Fig. 8 output of Discover node

B. Get data output

Here, read the data from the slave device. In that first slave device read the value from connected sensor and transmit to the master device. Master read data from slave and store it. The fig.9 shows the output of get data. The output shows the slave IP address and analog value of connected sensor. Slave device address 1 shows data: 32.4 which is temperature measured by PT100 in degree Celsius and salve device address 2 shows data: 1.7 which is current in Ampere measured by using ACS712 current sensor.



```
Discover Node = 1
Get Data = 2
Enter command number: 2
.....Get data from slave device....
address: 1
data: 32.4
*******************
address: 2
data: 1.7
*******************
```

Fig.9 Output of Get data

VI. CONCLUSION

The paper describes, design and implementation of I2C protocol. It is a master slave topology. It is designed for the distributed network that is one master and multiple slaves are connected to the master. We can connect up to 127 slave devices on I2C bus. Slave device read the data from PT100 sensor and ASC712 current sensor. Then slave transmit the data to the master device. Master read the data and store it.The overall system is low cost, consumes low power and more reliable.

REFERENCES

[1] Anagha A, M. Mathurakani, “ Prototyping of dual master I2C bu controller,” IEEE conference on communication and signal processing, April6-8,2016.

[2] Jacob Maxa, Thilo Kranchenfels, Helmut Beikirch, “ Near field communication interface for a packet based serial data transmission using dual interface EEPROM,” IEEE 20th conference on Emerging technologies and factory automation, September, 2015.

[3] Aruna Kommu, Raghavendra Rao Kanchi, “ Design a learning platform for the implementation of serial standars using ARM microconatroller LPC2148,” IEEE international conference on Recent Advances and Innovations in engineering, May 2014.

[4] Yassine Bouterra, Alaa Chabir, Asma Ben Mansour, Jawhar Ghommom, “ Development and implementation of real time system for distributed control of laboratry robot,” IEEE 11th International multi-conference on system, signal and devices, Feb 2014.

[5] The I2C bus specification version 2.1, January2010.

[6] UM10204, I2C bus specifiaction and user manul, Rev.6 ,4 April 2014

[7] ATMEGA328 datasheet.

[8] Datasheet of PT100.

[9] ACS712 current sensor datasheet.