

SERIAL PERIPHERAL INTERFACE (SPI) COMMUNICATION APPLICATION AS OUTPUT PIN EXPANSION IN ARDUINO UNO

Farid Baskoro¹, Miftahur Rohman², Aristyawan Putra Nurdiansyah³
1, 2, 3 Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Surabaya
Email: faridbaskoro@unesa.ac.id
Email: miftahurrohman@unesa.ac.id
Email: aristyawanputra03@gmail.com

Abstract - Serial Peripheral Interface (SPI) is a synchronous serial communication whose data or signal transmission involves Chip Select (CS) or Slave Select (SS) pins, Serial Clock (SCK), Master Out Slave In (MOSI), and Master In Slave Out (MISO). In the Arduino Uno, there are four pins that allow Arduino Uno to perform SPI communication. In this research, SPI communication is implemented to expand the output of the Arduino Uno by using the features of the MCP23S17 IC so that the Arduino Uno, which initially has 20 output pins, can expand to 36 output pins. The results of the research show that the Arduino Uno manages to control 36 output pins. 16 output pins from the MCP23S17, 16 output pins from the Arduino Uno, and 4 pins are used for the SPI communication line. The results of this study also show the form of the SPI communication signal from Arduino Uno in declaring 21 registers on MCP23S17, declaring the MCP23S17 pin register as output, and implementing the output using LEDs.

Keywords: SPI, Expansion, MCP23S17

I. INTRODUCTION

In the science of telecommunications and data delivery, there is a type of communication called serial communication. Komunikasi series consists of a synchronous series and asynchronous. Synchronous serial communication consists of several kinds, one of which is the serial peripheral interface or commonly abbreviated as SPI [1]. The first thing to note in conducting communication is the communication line [2].

Serial Peripheral Interface (SPI) is a communication whose communication line involves Pin Chip Select (CS) or Slave Select (SS), Serial Clock (SCK), Master Out Slave In (MOSI), and Master In Slave Out (MISO) [3]. SPI can communicate with f duplex and full-duplex methods [4]. SPI communication consists of masters and slaves [5], the master provides SCK to synchronize.

SCK signal in this SPI that controls when SPI changes data or read data. PIN CS or SS serves to determine the slave state in an active or inactive state. If the slave is active, then this indicates that the master wants to start the data exchange. The data exchange in question is the master conducts communication in the form of sending data as a command for slaves, or reading data as information from slaves. If the slave is in a non-active state, then the master will not be able to exchange data. CS pins can also be used as slave address selection which will exchange data with the master [6] Pin MOSI and MISO are used as pins sending register data information from master to slave and slave to master. One of ic (Integrated Circuit) that utilizes SPI as its communication line is IC MCP23S17

IC MCP23S17 is an IC that serves as Input or Output (I/O) expander. This IC is very helpful Arduino Uno which only has an output of 20 pins. One IC MCP23S17 has an output of 16 pins. So, with the limitations of pins on Arduino Uno, the author will utilize spi communication from IC MCP23S17 in expanding the pin on Arduino Uno, so that every one Arduino Uno is at least able to control 36 pin outputs.

II. LITERATURE REVIEW

MCP23S17 is a slave SPI device IC with a function as parallel I/O expansion using SPI as its communication line. This IC has a 28 pin configuration in table 1.

Table 1. 28 pin configuration MCP23S17

Pin Name	Pin Number	Pin Type
GPB0	1	I/O
GPB1	2	I/O
GPB2	3	I/O
GPB3	4	I/O
GPB4	5	I/O
GPB5	6	I/O
GPB6	7	I/O
GPB7	8	I/O
VDD	9	Power
VSS	10	Power
CS	11	I
SCK	12	I
SI	13	I/O
SO	14	O
A0	15	Address Pin

Pin Name	Pin Number	Pin Type
A1	16	Address Pin
A2	17	Address Pin
RST	18	I
INTB	19	O
INTA	20	O
GPA0	21	I/O
GPA1	22	I/O
GPA2	23	I/O
GPA3	24	I/O
GPA4	25	I/O
GPA5	26	I/O
GPA6	27	I/O
GPA7	28	I/O

MCP23S17 consists of several configurations of 8-bit registers to control input and output. Master can enable I/O on MCP23S17 as input or output by writing configuration bits (IODIRA/B). Then data for each I/O withheld or stored by the I/O register. MCP23S17 can be configured to operate in 8 bit or 16 bit mode via IOCON register. THE BANK.

IC MCP23S17 operation begins by lowering the CS or SS pin, from HIGH to LOW. Then on the master send slave address containing 4 bits definite and 3 bit slave address used.

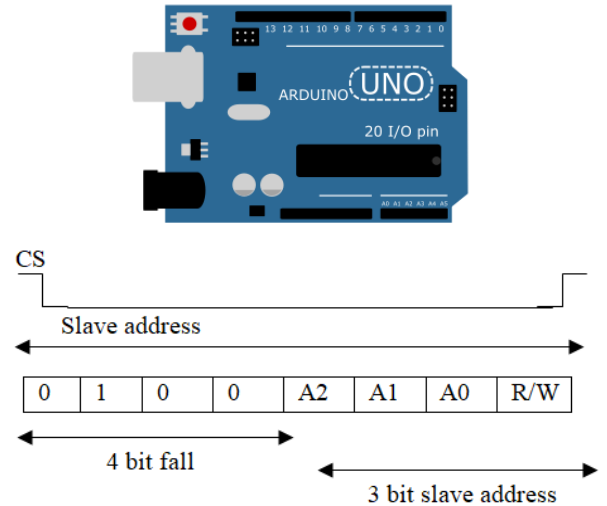
MCP23S17 has 21 configurations and register controls in table 2. 10 registers are addressed to PORTA and 10 registers are addressed to PORTB. 1 register (IOCON) is divided between two ports [7].

Table 2. Value of configuration and register MCP23S17

Register Name	Value
IODIRA	1111 1111
IODIRB	1111 1111
IPOLA	0000 0000
IPOLB	0000 0000
GPINTENA	0000 0000
GPINTENB	0000 0000
DEFVALA	0000 0000
DEFVALB	0000 0000
INTCONA	0000 0000
INTCONB	0000 0000
IOCON	0000 0000
IOCON	0000 0000
GPPUA	0000 0000
GPPUB	0000 0000
INTFA	0000 0000
INTFB	0000 0000
INTCAPA	0000 0000
INTCAPB	0000 0000
GPIOA	0000 0000
GPIOB	0000 0000
OLATA	0000 0000
OLATB	0000 0000

Microcontrollers are the most important electronic devices and control systems in communication [8]. One of the microcontrollers that are currently famous in the market is Arduino UNO. Arduino UNO is an open-source platform for electronic devices [9].

Arduino UNO has a board dimension of 70x53mm. Arduino UNO can utilize a USB connector as its voltage source [10]. Board Arduino Uno has atmega IC 328p, crystals of 16Mhz, and has 20 I/O. Arduino UNO can be used as a master in SPI communication, because it has PIN CS, SCK, MOSI, and MISO. Arduino's physical form can be seen in figure 1.



*R/W = 0 = write
*R/W = 1 = read

Figure 1. Arduino Uno and Slave Address IC MCP 23S17

In checking the I/O pin on the microcontroller, it can be done by connecting the output pin and LED (Light Emitting Diode). The output pin is connected with an LED anode, and ground with LED cathode. When the LED is on, it can be interpreted that the output pin on the microcontroller works.

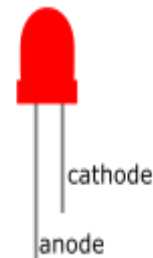


Figure 2. LED Pin out

III. METHOD

The design phase of this research begins with literature studies. The first literature study was to study previous journals on SPI communication from domestic and foreign sources. Next, learn about how to apply SPI on IC MCP23S17 with Arduino Uno from the MCP23S17 datasheet that has been published by Microchip. After the literature study has been completed, the next step is to design the system design.

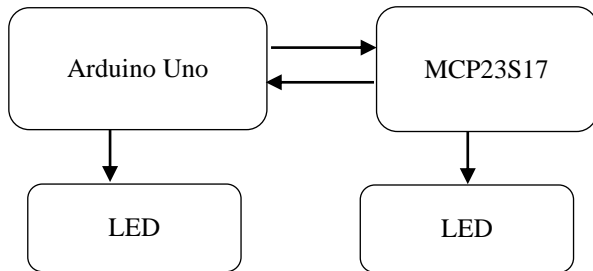


Figure 3. System Design Diagram Blocks

Figure 3 shows the block diagram of the system design. In the Arduino uno diagram block has an arrow direction to MCP23S17, this indicates that Arduino is used as a master, the master function here is the commander to MCP23S17 or the recipient of information from MCP23S17 by utilizing SPI communion. MCP23S17 works as a slave. The slave will work when the master instructs. When Arduino Uno orders THE MCP23S17 to activate its output, the LED will light up as commanded.

IV. RESULTS AND DISCUSSIONS

The results and discussion in this study are divided into several parts, the first part is the form of SPI signal on 21 registers, the second part is the form of SPI signal when activating GPA, the third part is b for SPI when activating GPB. The fourth part is the result of the LED flame on all output pins.

SPI Communication Signal Test Results on 21 registers

When using MCP23S17, the first thing to do is declare the value of 21 registers, so that MCP23S17 can work as instructed by Arduino. The form of SPI communication signal when sending the value of 21 registers can be seen in figure 4.

In figure 4, there is a display of spi communication signal form consisting of 3 constituent signals with oscilloscope settings of 5volt/div and 5us/div. The 3 signals have different color and shape displays. The signal is yellow, which is a signal issued by the CS pin.

A blue signal is a signal issued by an SCK pin, and a pink signal is a signal issued by a MOSI pin. On the SCK signal, there are 22 HIGH signals. Signal with writing is a slave address signal, then the signal with the writing 0-22

21 signal carrier SCK accompanying the delivery of the value of 21 registers on the PIN MOSI.

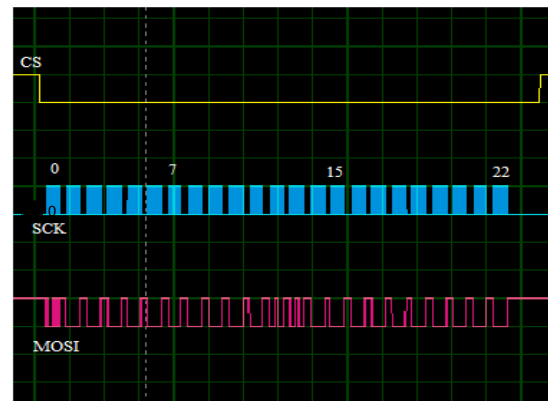


Figure 4 SPI Communication Signals

As explained in the library study, the MCP23S17 work operation begins with lowering the CS pin. After the CS pin becomes LOW, the next step is to send information or commands from Arduino to MCP23S17. This form of information signal or command is a value of 21 registers issued by MOSI pin from Arduino, then along with each register value sent, SCK pin also sends 16-bit signal as accompaniment.

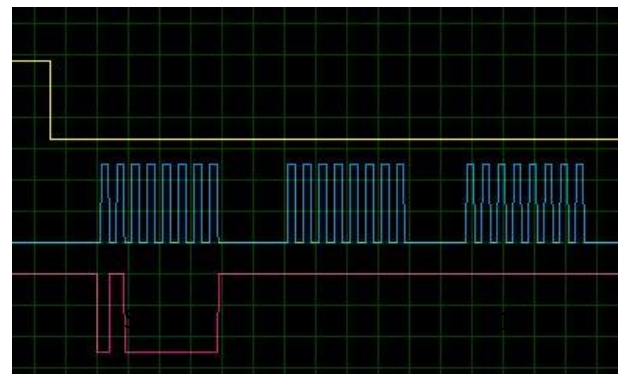


Figure 5. Signal Form of IODIRA and IODIRB Register Values

To see the signal shape of each register, the oscilloscope display in figure 5 is changed to 2volt/div and 0.5 us/div. In figure 5, there are 3 forms of signals on the PINK MOSI pin. The signal with the sa (slave address) text, has a decimal value of 64 or 0100000 in binary. Signal number 0, has a decimal value of 255 or 11111111 in binary. Signal number 1, has a decimal value of 255 or 11111111 in binary.

In each information delivery, Arduino's pink MOSI pin begins by sending a value of 64, decimal value 64 is the value used to describe the slave address. If you want to send information to the first MCP23S17, then Arduino sends a value of 64. If you want to send information to the second MCP23S17, Arduino sends a value of 66. If you want to send information to the third MCP23S17, then Arduino

sends a value of 68. This is the same as written in the study of the library challenge slave address.

After the first slave address value is sent, the next value is the value that starts the 21 registers i.e. IODIRA. IODIRA has a score of 255. Then the next value is dari IODIRB, IODIRB also has a value of 255, therefore, the pink signal in figures 5 number 0 and 1 increases after SA.

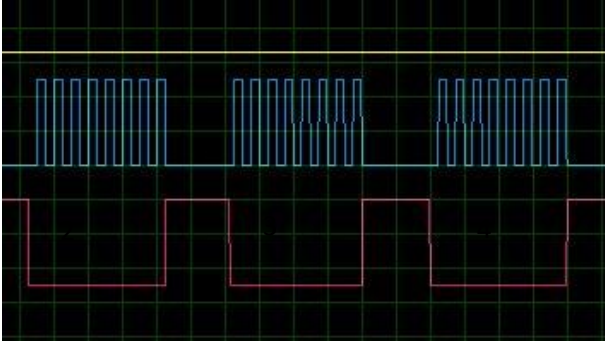


Figure 6. Signal form of IPOLA, IPOLB, and GPINTENA register value

The next registers are IPOLA, IPOLB, and GPINTENA. These three registers have a value of 0, so when the SCK pin sends its companion signal, the MOSI pin emits a LOW signal.

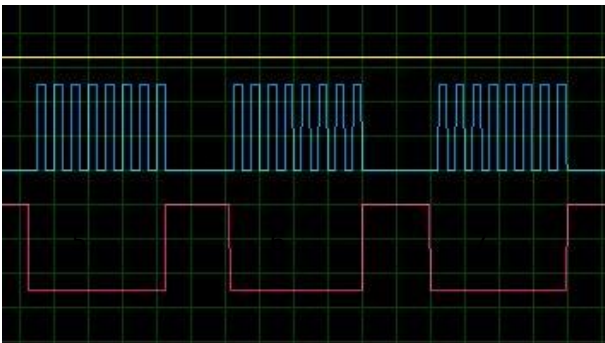


Figure 7. Signal form of GPINTENB, DEVALA, and DEVALB register values

The next registers are GPINTENB, DEVALA, and DEVALB. These three registers also have a value of 0, so that when the SCK pin sends its companion signal, the MOSI pin emits a LOW signal

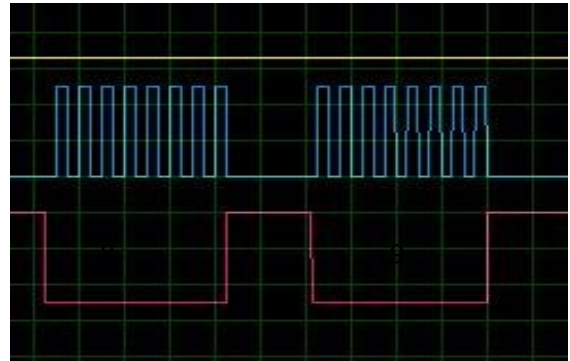


Figure 8. Signal form of INTCONA and INTCONB register values

The next registers are INTCONA and INTCONB. Both registers also have a value of 0, so when the SCK pin sends its companion signal, the MOSI pin emits a LOW signal.

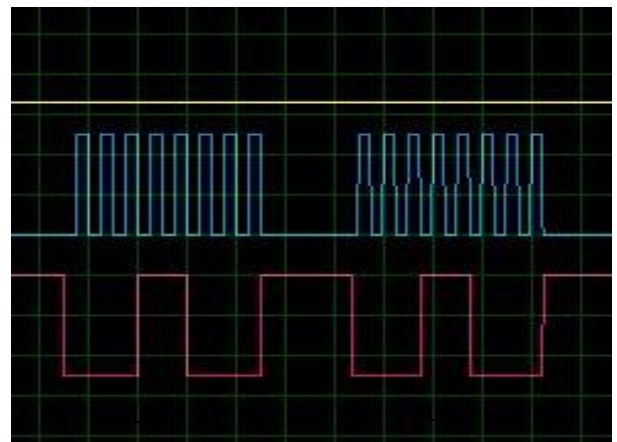


Figure 9 Signal form of IOCON register value

The 10th and 11th registers are IOCON. IOCON has a value of 24, so when the SCK pin sends its accompaniment signal, the MOSI pin not only emits a LOW signal. For registers 12 through 22, have a value of 0, so the shape is equal to figure 6, figure 7, and figure 8.

SPI Signal Test Results when activating GPA

Pinouts on MCP23S17 can be enabled as input or output pins. Inactivating the output on the MCP23S17, 2 delivery steps must be done by Arduino UNO. The first step is to send a register to declare the pin used in figure10. Figure 10 shows the shape of the signal that serves as a declaration of GPA0 as an output. SA has a decimal value of 64 because Arduino sends the command to the first MCP23S17. Signal 1 has a decimal value of 0, and the 2nd signal has a decimal value of 254.

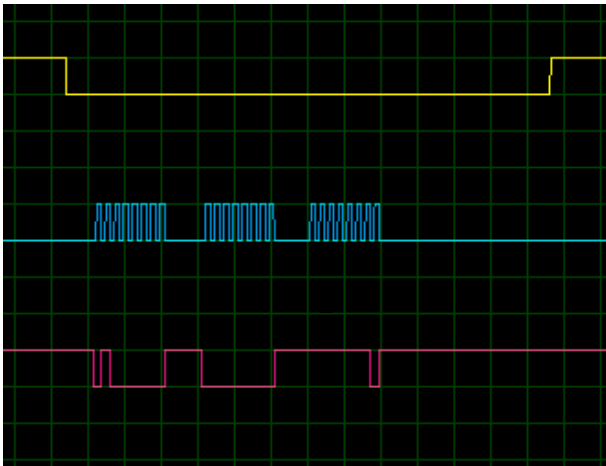


Figure 10. GPA0 Declaration Signal Form as Output

After sending the command in accordance with figure 10, the second step is Arduino uno send another command to determine the HIGH or LOW condition on the GPA0 output in the first MCP23S17.

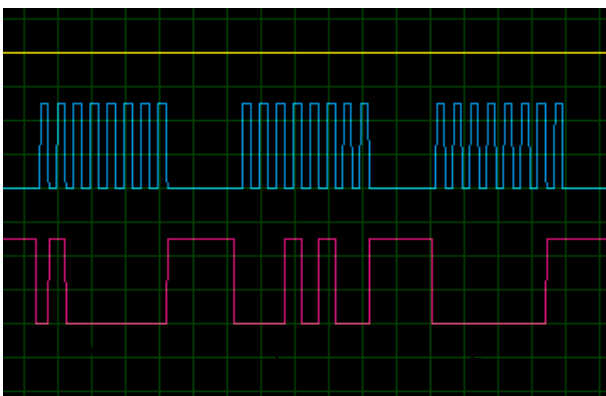


Figure 11. High GPA0 Condition Signal Form

Sinyal of MISO pin in figure 11 has SA with the decimal value of 64 because Arduino sends a command to the first MCP23S17. Signal 1 has a decimal value of 20, and signal 2 has a decimal value of 1. After Arduino sends both commands such as figures 10 and 11, the output on the GPA0 is high.

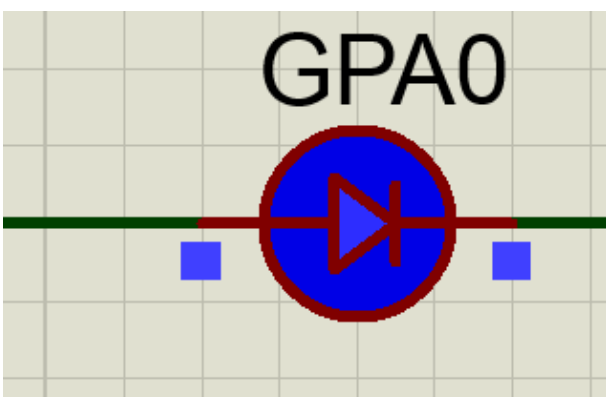


Figure 12. HIGH Condition GPA0 Output

If you want to change the condition of GPA0 to LOW, then the command in figure 11 is changed to as in figure 13, and the result can be seen in figure 14.

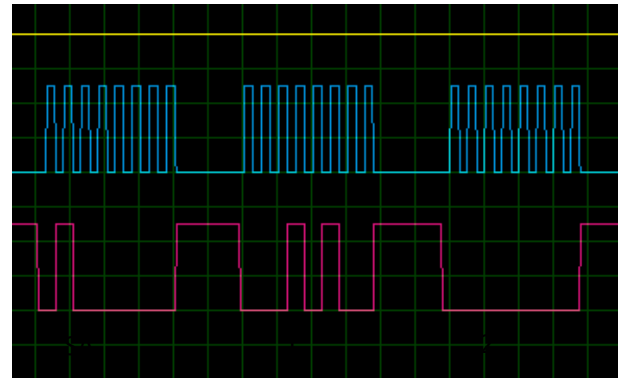


Figure 13. Low GPA0 Condition Signal Shape

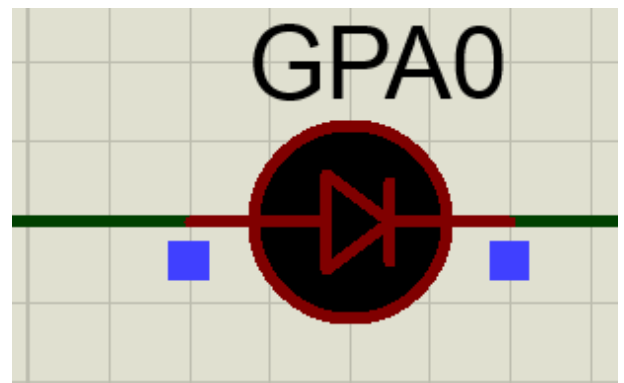


Figure 14. GPA0 Output Low Condition

SPI Signal Test Results when activating GPB

How to activate GPB pin almost same as GPA, Arduino UNO must send 2 commands first. However, there is a difference in decimal values in GPB. If the MOSI signal declaration numbered 1 in GPA is 0, then the MOSI signal declaration with the 1st name in GPB is 1 and the 2nd signal has a decimal value of 254 as shown in figure 15.

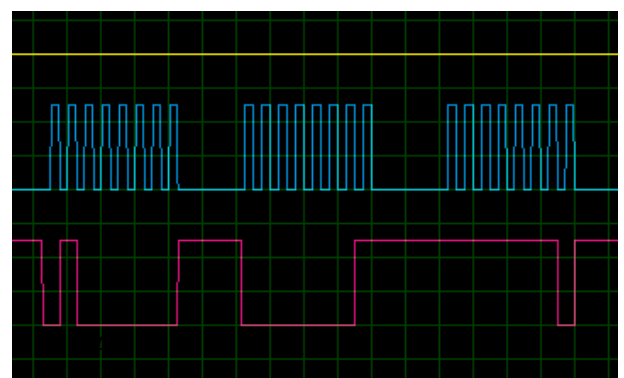


Figure 15. GPB0 Declaration Signal Form as Output

After Arduino sends GPB0 declaration command like figure 15. The second step is to send the register value to determine the condition HIGH or LOW. MISO pin with SA writing has a decimal value of 64. The signal with the number 1 is decimal to 21. The signal with the number 2 has a decimal value 1 as in figure 16.

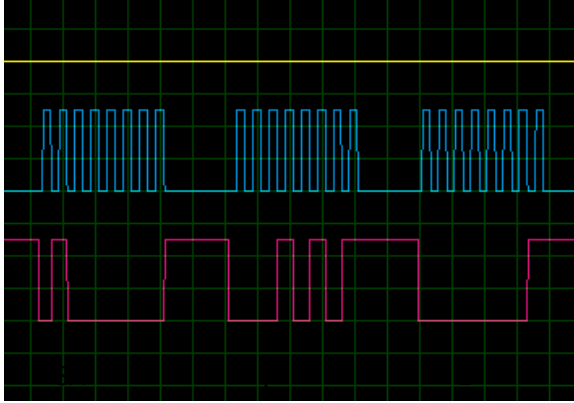


Figure 16. High GPB0 Condition Signal Form

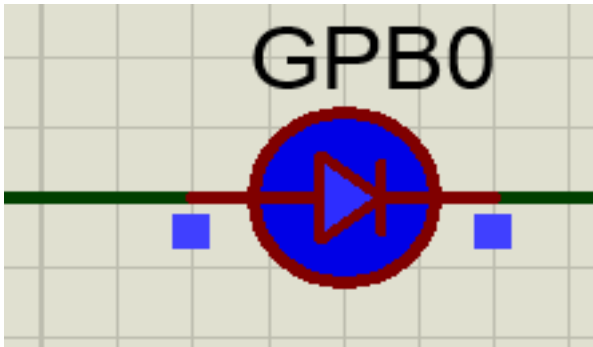


Figure 17. GPB0 Output High Condition

Figure 17 shows GPB0 in HIGH condition after Arduino UNO sends 2 commands such as figure 15 and 16. If you want to change the condition of GPB0 to LOW, then the command in figure 16 is changed to as in figure 18, and the result can be seen in figure 19.

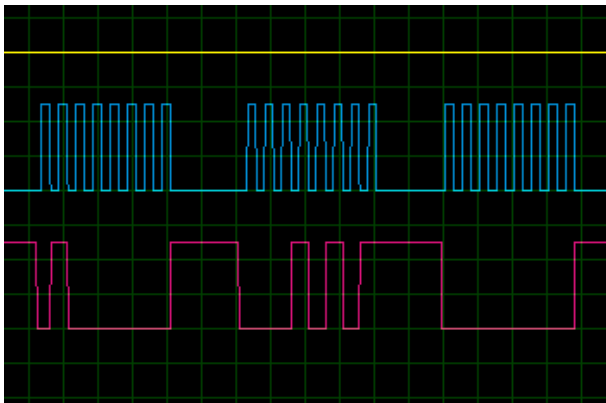


Figure 18. Low GPB0 Condition Signal Shape

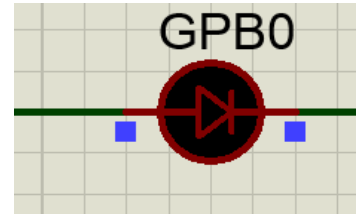


Figure 19. GPB0 Output Low Condition

LED Test Results on Output Pins

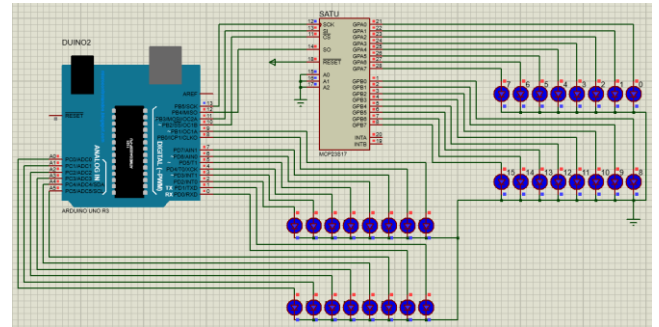


Figure 20. Display Of High Output Pin Conditions

Figure 20 is a simulation of 16 pins of MCP23S17 output and 16 pins of Arduino Uno output in HIGH condition, 4 pins of Arduino Uno used as SPI communication line. The top LED line is 8 GPA pins and the bottom LED line is 8 GPB. Pins A2, A1, and A0 are connected to the ground to describe the bit value of 000 in the first MCP23S17 so that the first slave address has a decimal value of 64.

V. CONCLUSION

From the research that has been done, it can be concluded that SPI communication on MCP23S17 as a slave and Arduino Uno as master successfully implemented for output pin expansion on Arduino Uno, by sending value to 21 register, then sending value to describe GPA pin or GPB as output, and sending value to determine output in HIGH or LOW condition. After the delivery of the value is successfully done, then Arduino Uno has successfully performed control over 36 output pins. 16 output pins from MCP23S17, 16 output pins from Arduino Uno, and 4 pins used for SPI communication lines.

REFERENCES

- [1] Susana, Ratna, Muhammad Ichwan, and Savero AL Phard. 2018. "Penerapan Metoda Serial Peripheral Interface (SPI) Pada Rancang Bangun Data Logger Berbasis SD Card." ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika 4(2): 208.
- [2] Puspitaningayu P, Baskoro F, Buditjahjanto IA. Komunikasi Data dan Komputer.
- [3] Trivedi, Dvijen, Aniruddha Khade, Kashish Jain, and Ruchira Jadhav. 2018. "SPI to I2C Protocol Conversion Using Verilog." Proceedings - 2018 4th International

- Conference on Computing, Communication Control and Automation, ICCUBEA 2018: 1–4.
- [4] Arshad, M. K.Md, U. Hashim, and Chew Ming Choo. 2006. “Characteristics of Serial Peripheral Interfaces (SPI) Timing Parameters for Optical Mouse Sensor.” IEEE International Conference on Semiconductor Electronics, Proceedings, ICSE: 576–82.
- [5] Anand, N., George Joseph, Suwin Sam Oommen, and R. Dhanabal. 2014. “Design and Implementation of a High-Speed Serial Peripheral Interface.” 2014 International Conference on Advances in Electrical Engineering, ICAEE 2014.
- [6] Li, Li Li, Jing Yu He, Yong Peng Zhao, and Jian Hong Yang. 2014. “Design of Microcontroller Standard SPI Interface.” Applied Mechanics and Materials 618(March 2017): 563–68.
- [7] Microchip. 2016. “Mcp23017/Mcp23S17.” : 1–42. [papers3://publication/uuid/FCEDDDC0-C695-4128-BF5A-A73A715E4E7F](https://www.microchip.com/publication/uuid/FCEDDDC0-C695-4128-BF5A-A73A715E4E7F).
- [8] Ferreira, Luis F., Emilio L. Matos, Luis M. Menendez, and Enrique Mandado. 2005. “MILES: A Microcontroller Learning System Combining Hardware and Software Tools.” Proceedings - Frontiers in Education Conference, FIE 2005: 7–11.
- [9] Banerjee, Soumyendu et al. 2019. “Real-Time Monitoring and Control of Consumed Power for Household Appliances Using Arduino Uno through Bluetooth and Android Application.” Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019 2019-April(Icoei): 529–33.
- [10] Kasenda, Sonny, Doostenreyk Kantohe, Maureen Langie, and Anthoinete Waroh. 2018. “Light Intensity Control Prototype Design Using Arduino Uno.” Proceedings - 2018 International Conference on Applied Science and Technology, iCAST 2018: 563–66.