



EVALUATION OF I2C COMMUNICATION PROTOCOL IN DEVELOPMENT OF MODULAR CONTROLLER BOARDS

Alvin Jacob, Wan Nurshazwani Wan Zakaria, and Mohd Razali Bin Md Tomari

Advanced Mechatronics Research Group, Faculty of Electrical & Electronic Engineering,

University Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, Johor, Malaysia

alvinjacob90@gmail.com

ABSTRACT

This study proposes the usage of I2C, (Inter-Integrated Circuit) communication protocol for a modular general purpose controller board of mobile robots. For the past few years, a trend of purchasing an off the shelf controller board by designers that would later be altered to fit a robot specification and design are has been increasing. This modular controller board allows users to use this modular controller board for their designs with minimal modification or without. Rather than purchasing an all-in-one controller board that might be costly, modularity means that only the needed modular board will be purchased. A total of four modular controller boards has been developed, which are main controller module, motor controller module, sensor module, pneumatic controller module, and one extension power supply module. All this modular controller boards incorporates the latest I2C communication protocol between the modular boards, which is faster and reduces the connecting wire between the modular boards. The potential of using I2C communication protocol in the proposed board is verified through a comprehensive series of experiments and the results suggest that the I2C is suitable and robust enough to be used in the development of this modular boards.

Keywords: I2C, modular controller board, communication system, mobile robot.

INTRODUCTION

Robots can be considered as machines that are capable of carrying out a series of actions automatically. Robots are a mechanical or virtual agent, usually a mixture of electronic controller boards and mechanical links that make up this electro-mechanical machine. Robots can be either autonomous or semi-autonomous and range from humanoids to simple robot that perform normal task ("Robot," 2006). Mobile robot is a subcategory under the vast types of robots, it is an automated machine that is capable to move in any given environment based on their respective locomotion method. Normally, robots are programmed to fulfil specific task based on the required application (Lazinica, 2006).

For instance in Asia, The Asia-Pacific Robot Contest (ABU Robocon) is an Asian Oceania College robot competition, where the applications of mobile robots are most apparent. In the competition, robots compete to complete a task within a set period of time ("ABU Robocon," 2002). In order for a robot to move, controller boards play an important role because it is used to control just about everything that is used to actuate the robot from reading a sensor input until activating a relay to move an output.

A controller board can be considered as an expansion card, or a stand-alone device that connects with a peripheral device. This controller board may be a link in between the main controller and the slave controller or a controller that is connected to an external device that manages the overall operation of the mobile robot (Jones, 2004). When linking main controller and a slave controller, some method of communication is required. Many controller board implement parallel communication where all the peripherals are connected using an amount of data lines depending on the data length, for example an 8-

bit data length will require an 8-bit data line for parallel transmission to occur. This is different with the serial communication where serial is a process of sending data sequentially over a communication channel, one bit at a time. Serial data transmission uses less data line which brings down the overall cost needed for it to be implemented. I2C (Inter-Integrated Circuit), is one of the popular serial data transfer protocol that is being used widely in many industries.

The advancement in circuit board technology has made way for designing controller board that are more complex and sophisticated in term of functionality. Thus, this project aims to implement I2C communication protocol for the communication between the developed modular controller boards as a general purpose controller board when connected to any mobile robot that needs to be controlled.

I2C COMMUNICATION PROTOCOL

Introduction of I2C

I2C or I²C is as pronounced I-squared-C, is a multi-master, multi-slave, single-ended, serial computer bus invented by Philips Semiconductor, known today as NXP Semiconductors, used for attaching low-speed peripherals to computer motherboards and embedded systems (NXP Semiconductors, 2014). I2C was originally developed in 1982 by Philips for various Philips chips. The original specification that the protocol was made for is only 100 kHz communications in which only provided 7-bit addresses. Thus, limiting the number of devices connected on the bus to 112. In 1992, the first public specification was published, communication speed was increased to 400 kHz fast-mode which offered 10-bit addresses. Later three additional modes were specified;



fast-mode plus at 1MHz, high-speed mode at 3.4MHz and ultra-fast mode at 5MHz (Corcoran, 2013).

Most modern microcontroller support I2C communication at 400 kHz. The I2C bus was initially designed to ease both systems designers and equipment manufacturers and to maximize hardware efficiency and circuit simplicity. All I2C-compatible devices incorporate an on-chip interface that allows them to communicate directly with each other via a simple two-wire bus. I2C communication uses only two bidirectional open-drain lines, Serial Data Line (SDA) and Serial Clock Line (SCL), pulled up with resistors, typical voltages used are +5 V or +3.3 V (NXP Semiconductors, 2014). Figure-1, illustrates a typical I2C Bus Connection.

The IC (Integrated Circuit) that initiates a data transfer on the I2C bus is considered the Bus Master. Consequently, at the same time, all the other ICs are considered to be Bus Slaves in which all have a 7-bits slave addresses. The data transferred between the lines are divided into 8-bit bytes, and a few control bits are added for controlling the communication start, end, direction and an acknowledgment mechanism (Corcoran, 2013). The maximum number of nodes connected to the I2C communication lines are simply limited by the address space available. The total bus capacitance of 400 pF will restrict the practical communication distances to a few meters. Although I2C is meant for short distance communication, virtually any number of slaves and masters can communicate with each other on these 2 signal.

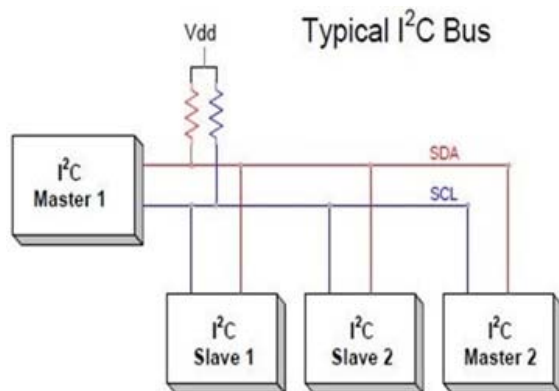


Figure-1. Typical I2C bus connection (Corcoran, 2013).

One of the most important design of the I2C bus design is its use of a simple open-drain architecture. This means that the SCL and SDA signals lines operate as pull-downs for the bus. Thus, the normal bus state is active-high, and it will be pulled down by these output pins. In normal operation, the SDA (Data line) only changes state during the low state of the SCL (Clock line).

During the high state of the SCL line, the high-to-low and low-to-high transitions of the SDA line indicates the start and stop conditions. Thus, this method eliminates extra control bit to be used to initiate and terminate data transfers on the bus. Figure-2 shows a complete data

transfer for I2C over the bus lines. Note that in the original reference design of U.S. patent 4,689,740, pull-up resistors are required between the open drain pin of the IC and the bus (NXP Semiconductors, 2014).

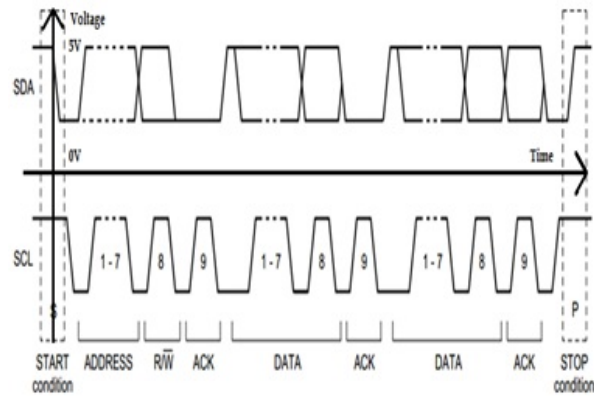


Figure-2. A complete I2C data transfer bit representation (NXP Semiconductors, 2014).

I2C integration into modular controller boards

A different approach has been used by Xavier Righetti, where the use of the popular and proven I2C protocol has been proposed in wearable architecture (Righetti & Thalmann, 2010). The goal is to develop a modular approach towards wearable computing in a sense that users simply attach specific modules to their garments depending on their requested functionalities. The I2C bus is embedded in the clothes and magnets are used as connectors to physically attach the modules and connect them at the same time to the bus (Righetti & Thalmann, 2010). This usable principle makes the clothes versatile as they can be entirely personalized in terms of functionality. These types of wearable electronics are growing rapidly as user want to do many thing on-the-go and this is the easiest way to do it.

According to NXP Semiconductors (NXP Semiconductors, 2014), I2C, is a multimaster serial single-ended computer bus invented by Philips which is used for attaching low-speed peripherals to a motherboard, embedded system, cell phone, or other electronic device. The I2C is not to be confused with the term Two Wire Interface which only describes a compatible hardware interface. One of the most significant advantages of using I2C is that the 2-wire connection allows for the communication between modular boards to be easily implement and also it is reduces the messy connection, but all this happen as long as the connected peripheral support I2C communication protocol. The main reason for using this communication protocol in this wearable design is the small amount of data lines required in total to connect all the sensors together.

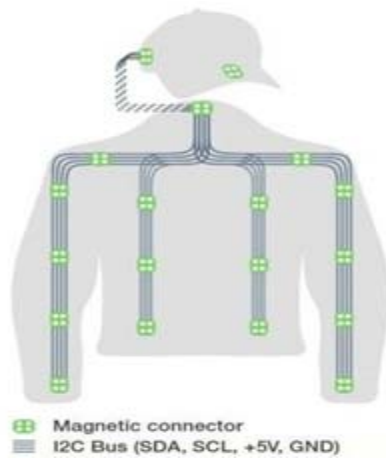


Figure-3. The internal connection and architecture of modular wearable framework (Righetti & Thalmann, 2010).

Figure-3 illustrates how the I2C is implemented in wearable architecture such as clothes. Wearable modules that could be attached to this standard bus could be classified in different categories: input, output, processing, communication, storage, and power modules. Additionally, a master module is required in order to setup the functionalities and to manage the communication between all the other slave modules. This is all possible by using the I2C protocol. Furthermore, only one pack of batteries is required instead of having a battery for each module, thus reducing the weight and size of each module. Another research used the I2C bus method to attach the sensors required in data collection to a microcontroller for a mobile measuring device that involves high speed data acquisition (Jacob, Wan Zakaria, & Md Tomari, 2015a).

MODULAR CONTROLLER BOARD

Modular controller board design

The main intention of this project is to integrate the I2C communication protocol into the development of the modular controller board. To implement the I2C communication protocol between these controller boards, Arduino development kit is used. For this, C language is used to program the Arduino board since the Arduino IDE uses it as the main programming language. I2C protocol is used and implemented for the communication between the main controller and slave modules. The usage of I2C bus simplifies connection between modules, thus reducing the connecting wires. The main controller caters for the wired PS2 (Play Station 2) controller and communicates through the I2C bus to the other modules, sending or receiving command. The software implementation for the I2C is a master-write-slave-read implementation; by using this master sends out command to specific addressed slave module. After receiving the command, the specified slave module performs the command.

Hardware configuration

The modular controller board can be divided into four main modules as follows;

- **Main controller:** comprises of main controller that controls the overall communication between the modular boards. This controller also is connected to an UART PS2 receiver that decodes the instruction given by the PS2 controller and sends it to the main controller.
- **Motor controller module:** this module is made up of a microcontroller that communicates with the main controller to control the motor based on the command received. Microcontroller is integrated on boards to control the operation of motor.
- **Sensor module:** this module is the main input module for the whole system; this module accommodate up to 16 digital inputs. The boards consist of a MCP23017 I2C port expander chip that act as a slave device.
- **Pneumatic controller module:** this module contains relays that are used to control the input signal to the pneumatic control valve. A total of eight outputs is provided to control 8 pneumatic valves via relays. This module has a MCP23008 I2C port expander chip to act as slave device on the I2C bus.

Figure-4 illustrates the connection between the main controller and the proposed modules. The overall hardware setup diagram and the operation of this system. The controller board is fabricated according to the design in Figure-4.

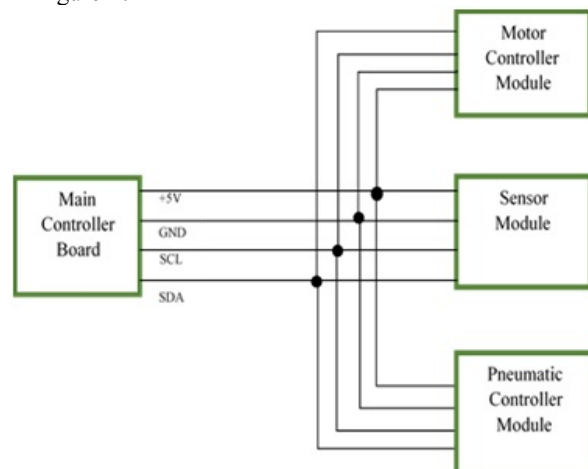


Figure-4. Illustration of the overall system.

Data acquisition

Figure-5 shows the data acquisition process initiated by the main controller board through a PS2 (Play Station 2) input controller. Then, the data being processed by the microprocessor and finally activates the output mechanism. First, the signal from the PS2 Controller is



converted into UART (Universal Asynchronous Receiver/Transmitter) and is send to the main controller.

The Arduino Uno microcontroller which performs the decoding and sending data to the slave device to read via the I2C bus. For the data acquisition part, the main program is written in C language and then is loaded into the Arduino Uno. The MCP23008 and MCP23017 I2C port expanders chip is used in the modular controller boards are 7-bit addressable and their working speed vary from 100 kHz, 400 kHz till 1.7 MHz.

Once the data is on the I2C bus, only the respective slave device can read the data. This is because all the slave devices have their own 7-bit addresses and the command that the master controller sends onto the I2C bus will contain the respective address to the slave devices. Thus, only the slave device with the address is able to read the data that has been sent by the master.

Lastly is when the slave device has received the command and decodes the command from the master and ready to execute the command.

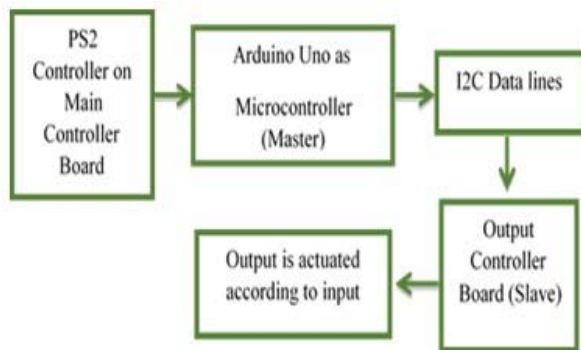


Figure-5. Flow chart diagram of the data acquisition.

RESULTS AND DISCUSSIONS

The performance of each modular board is tested based on the total time taken to execute a command referring to the flow chart diagram in Figure 5. The Cycle time is the period required to complete one cycle of an operation; or the time taken to complete a task from the beginning till the end.

Before testing every modular controller board the cycle time of the microcontroller used, Arduino and the I2C port expanders chip used is calculated by the time it takes to execute a command. The cycle time is calculates using equation (1).

$$\frac{1}{\text{Frequency}} = \text{Time} \quad (1)$$

$$\frac{1}{6\text{MHz}} = 62.5\text{ns} / 1 \text{ instruction} \quad (2)$$

$$\frac{1}{1.7\text{MHz}} = 588 \text{ ns} / 1 \text{ instruction} \quad (3)$$

According to equation (2), it can be said that the Arduino takes 62.5ns to execute one instruction. Whereas, the speed where the I2C chip operates is at 588ns as according to equation (3). By using the I2C communication protocol the overall connection wires between the modular controllers boards have been reduced, this is because only two lines are used for communication. This also decreases the amount of data processing that is done by the microcontroller, thus reducing the computational load since operations are done in parallel over different modular boards. The operation of the controller boards has been tested for its capability to perform two different task simultaneously.

Modular board cycle time

The cycle time for each modular controller board is evaluated and is recorded into Table-1, using the values from equation (2) and (3).

Table-1. Total cycle time for individual modular controller board.

Modular Controller Board	Cycle Time (Seconds)	Operations Executed
Main Module	Arduino Process time = <u>250 ns</u> (4 command x 62.5 ns) PS2 receiver = <u>200 ns</u> (PIC running on a 20 MHz crystal / 4) Total Time = <u>450 ns</u>	1) Receive information from the PS2 Controller 2) Convert the control signal into commands 3) Send command signal to slave device
Motor Module	Main module processing time = <u>450 ns</u> Motor module controller = <u>250 ns</u> (4 command x 62.5 ns) RUN instruction = <u>10 ms</u> (time taken by the motor driver) CW/CCW instruction = <u>10 ms</u> STOP instruction = <u>10 ms</u> Total Time = <u>30.0007 ms</u>	1) Receive information from the main (master) controller or sensor module 2) Convert the received control signal into motor control commands 3) Execute the motor control command



Sensor Module	Main module processing time = <u>450 ns</u> Pneumatic Module Processing time = <u>5 ms</u> MCP23017 processing time = <u>588 ns</u> START time = <u>0.32 μs</u> DATA time = <u>0.01 μs</u> STOP time = <u>0.16 μs</u> Total Time = <u>5.001498 ms</u>	1) Receive information from sensors connected. 2) Receive read instruction from the main (master) controller 3) Send the sensor information to main controller
Pneumatic Module	Main module processing time = <u>450 ns</u> Relay reaction = <u>5 ms</u> Valve reaction = <u>5 ms</u> MCP23008 processing time = <u>588 ns</u> START time = <u>0.32 μs</u> DATA time = <u>0.06 μs</u> STOP time = <u>0.16 μs</u> Total Time = <u>10.001548 ms</u>	1) Receive control command from the main (master) controller or sensor module 2) Execute the control command received

System cycle time

The performance of the developed modular controller boards was tested when all the modular board are connected together based on two different programming method, sequential programming and pooling programming. Table-2 shows the result for the 4 different test scenarios that were conducted, which are

- Run 4 DC brushless motor connected to the motor controller board when then button on the PS2 controller is activated
- On and off 4 pneumatic valves connected to the pneumatic controller board when the input button on the PS2 controller is activated
- Run 4 DC brushless motor connected to the motor controller board when the sensor 1 connected to the sensor board is activated
- On and off 2 pneumatic valves connected to the pneumatic controller board when the sensor 1 and sensor 2 connected to the sensor board is activated

Table-2. Total cycle time for overall modular controller board operation.

Scenario A:	Scenario B:
Main controller = <u>450 ns</u>	Main controller = <u>450 ns</u>
Motor Controller Board = <u>30 ms</u>	Pneumatic Controller Board = <u>10ms</u>
Total Time = <u>30.00045ms</u>	Total Time = <u>10.00045 ms</u>
Scenario C:	Scenario D:
Main controller = <u>450 ns</u>	Main controller = <u>450 ns</u>
Sensor Controller Board = <u>5 ms</u>	Sensor Controller Board = <u>5 ms</u>
Motor Controller Board = <u>30 ms</u>	Pneumatic Controller Board = <u>10 ms</u>
Total Time = <u>35.00045ms</u>	Total Time = <u>15.00045ms</u>

The next test that was conducted is to test the total time taken for the execution of two different type of programming method used to program tasks that is needed to be performed. Table-3 shows the result for a simple scenario, which is

- Run 1 DC motor connected to the motor controller board and 1 pneumatic valve connected to the pneumatic controller board simultaneously when the input push button connected to the sensor board is pressed

Table-3. Total cycle time for different programming method.

Scenario E:	
Sequential Programming	Pooling Programming
Main controller = <u>450 ns</u>	Main controller = <u>450 ns</u>
Sensor Controller Board = <u>5 ms</u>	Sensor Controller Board = <u>5 ms</u>
Motor Controller Board = <u>30 ms</u>	Motor Controller Board = <u>30 ms</u>
Pneumatic Controller Board = <u>10 ms</u>	Pneumatic Controller Board = <u>10 ms</u>
Total Time = <u>45.00045ms</u>	Total Time = <u>35.00045ms</u> (pneumatic executes faster than motor)



From this test and results, the pooling based programming is faster because it can perform 2 tasks in parallel rather than 1 by 1 execution, a similar test conducted by (Jacob, Wan Zakaria, & Md Tomari, 2015b) in their mobile measuring device suggest that the pooling based programming method is best used if there is a need to read more than one sensor. By examining the results, it can be said that the concept of using pooling programming is more efficient for this project because it allows the modular board to perform multitasking and it does not waste precious time by making the module board waiting for commands and looping.

CONCLUSIONS

As a conclusion, after all the evaluation test that has been done it can be said that the I2C protocol is a robust and flexible communication protocol and is very suitable to be used as the main communication protocol in the development of modular controller boards. The success of completing the integrating of I2C communication protocol into depended on two main factor, the integrity of the electronic design and the simplicity if the software design.

The general purpose modular board development was based on current and important requirement in mobile robot technology. The need for a well working and yet reliable controller board is in demand to the robotic industries. Much research has been put into finding the right component to construct the PCB (Printed Circuit Board). The usage of I2C bus has greatly simplify the PCB design, this is because only two data line is needed for communication.

The development of the software algorithm for controlling the overall behaviour can be improved with all the latest technology in code optimizing, this will greatly contribute to a faster respond in the control system. This statement is supported by the results of the test conducted, where it can be said that the pooling based programming method is more efficient than the sequential based programming method for this modular controller board.

ACKNOWLEDGEMENTS

This Research was conducted under the supervision of Dr. Wan Nurshazwani Wan Zakaria and is supported by Universiti Tun Hussein Onn Malaysia.

REFERENCES

- [1] ABU Robocon. (2002). Retrieved June 12, 2015, from http://en.wikipedia.org/wiki/ABU_Robocon
- [2] Corcoran, P. (2013). Two Wires and 30 Years: A Tribute and Introductory Tutorial to the I2C Two-Wire Bus. *Consumer Electronics Magazine, IEEE*, 2(July), 30–36. <http://doi.org/10.1109/MCE.2013.2257289>
- [3] Jacob, A., Wan Zakaria, W. N., & Md Tomari, M. R. (2015a). Implementation of Bluetooth Communication in Developing a Mobile Measuring Device to Measure Human Finger Movement. In *International Conference on Electrical and Electronic Engineering 2015 (IC3E 2015)*. Melaka, Malaysia.
- [4] Jacob, A., Wan Zakaria, W. N., & Md Tomari, M. R. (2015b). Quantitative Analysis of Hand Movement in Badminton. In *2015 2nd International Conference on Computer and Communication Engineering (ICOCOE'2015)*. Phuket, Thailand: Springer.
- [5] Jones, D. (2004). PCB Design Tutorial. June 29th. Retrieved from www.alternatezone.com
- [6] Lazinica, A. (2006). *Mobile Robots: Towards New Application*. (A. Lazinica, Ed.) Advanced Robotic Systems International.
- [7] NXP Semiconductors. (2014). UM10204 I2C-bus specification and user manual. Retrieved from www.nxp.com
- [8] Righetti, X., & Thalmann, D. (2010). Proposition of a modular I2C-based wearable architecture. *Melecon 2010 - 2010 15th IEEE Mediterranean Electrotechnical Conference*, 802–805. <http://doi.org/10.1109/MELCON.2010.5475965>
- [9] Robot. (2006). Retrieved May 11, 2015, from <http://www.merriam-webster.com/dictionary/robot>