10

Development of Mobile Robot Based on I^2C Bus System

Surachai Panich
Srinakharinwirot University
Thailand

1. Introduction

Mobile robots are widely researched in many applications and almost every major university has labs on mobile robot research. Also mobile robots are found in industry, military and security environments. For entertainment they appear as consumer services. They are most generally wheeled, but legged robots are more available in many applications too. Mobile robots have ability to move around in their environment. Mobile robot researches as autonomously guided robot use some information about its current location from sensors to reach goals. The current position of mobile robot can be calculated by using sensors such motor encoders, vision, Stereopsis, lasers and global positioning systems. Engineering and computer science are core elements of mobile robot, obviously, but when questions of intelligent behavior arise, artificial intelligence, cognitive science, psychology and philosophy offer hypotheses and answers. Analysis of system components, for example through error calculations, statistical evaluations etc. are the domain of mathematics, and regarding the analysis of whole systems physics proposes explanations, for example through chaos theory.

This book chapter focuses on mobile robot system. Specifically, building a working mobile robot generally requires the knowledge of electronic, electrical, mechanical, and computer software technology. In this book chapter all aspects of mobile robot are deeply explained, such as software and hardware design and technique of data communication.

2. History of mobile robots (Wikipedia, 2011)

During World War II, the first mobile robots emerged as a result of technical advances on a number of relatively new research fields like computer science and cybernetics. W. Grey Walter constructed Elmer and Elsie, which were equipped with a light sensor. If they found a light source, they would move towards it, avoiding or moving obstacles. The Johns Hopkins University develops mobile robot named Beast. Beast used sonar to move around. Mowbot was the first automatically mobile robot to mow the lawn. The Stanford Cart for line follower was a mobile robot, which can follow a white line by using a camera. The mobile robot is developed to navigate its way through obstacle courses and make maps of its environment. The Stanford Research Institute researched on Shakey mobile robot. Shakey had a camera, a rangefinder, bump sensors and a radio link. The Soviet Union explores the surface of the moon with Lunokhod 1, a lunar rover as shown in Fig.1.

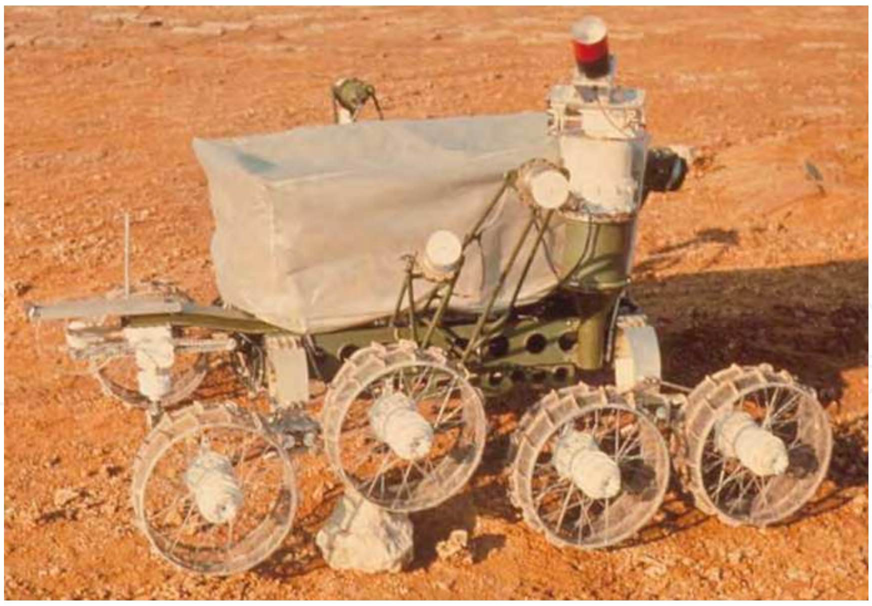


Fig. 1. A model of the Soviet Lunokhod-1 Moon rover released by the Science Photo Library

Fig. 2. Helpmate, autonomous mobile hospital robots

The team of Ernst Dickmanns at Bundeswehr University Munich built the first robot cars, driving up to 55 mph on empty streets. The Hughes Research Laboratories demonstrated the first cross-country map and sensor-based autonomous robotic vehicle. Mark Tilden invented BEAM robotics. Joseph Engelberger worked with colleagues to design the first commercially available autonomous mobile hospital robots named Helpmate as shown in Fig.2. The US Department of Defense funds the MDARS-I project for the Cyber-motion indoor security robot. Edo Franzi, André Guignard and Francesco Mondada developed Khepera, an autonomous small mobile robot. Dante I and Dante II as shown in Fig.3, walking robots used to explore live volcanoes were developed by Carnegie Mellon University.

Fig. 3. Dante II developed by Carnegie Mellon University

The twin robot vehicles VaMP and VITA-2 of Daimler-Benz and Ernst Dickmanns of UniBwM drive more than one thousand kilometers on a Paris three-lane highway in standard heavy traffic at speeds up to 130 km/h.

Fig. 4. Sojourner Rover, NASA

Semi-autonomous ALVINN steered a car coast-to-coast under computer control for all but about 50 of the 2850 miles. Throttle and brakes, however, were controlled by a human driver. The Pioneer programmable mobile robot becomes commercially available at an affordable price, enabling a widespread increase in robotics research and university study over the next decade as mobile robotics becomes a standard part of the university curriculum. NASA sends the Mars Pathfinder with its rover Sojourner to Mars as shown in Fig.4. The rover explores the surface, commanded from earth. Sojourner was equipped with a hazard avoidance system. Sony introduces Aibo as shown in Fig.5, a robotic dog capable of seeing, walking and interacting with its environment.

Fig. 5. Aibo, **Sony**

The PackBot remote-controlled military mobile robot is introduced as shown in Fig.6. PackBot is current base model using a videogame style hand controller to make easy control. PackBot is designed for improvised explosive device identification and disposal, infantry troops tasked with improvised explosive device inspection, help SWAT teams and other first responders with situational awareness.

Fig. 6. PackBot demonstrated by the French military

Swarm bots resemble insect colonies as shown in Fig.7. Typically they consist of a large number of individual simple robots that can interact with each other and together perform complex tasks.
Robosapien, a biomorphic commercially available robot designed by Mark Tilden as shown in Fig.8. The autonomous robots work together to make a map of an unknown environment and search for objects within the environment.

Fig. 7. Swarm bots: insect colonies behavior

Fig. 8. Robosapien designed by Mark Tilden

Sony introduced an autonomous service robot system named a lower-cost PatrolBot as shown in Fig. 9. The mobile robot becomes continue commercial product.

Fig. 9. PatrolBot introduced by Sony

The Tug as shown in Fig.10 becomes a popular means for hospitals to move large cabinets of stock from place to place for carrying blood and other patient samples from nurse stations to various labs.

Fig. 10. The Tug, Aethon's Automated Robotic Delivery System

Fig. 11. BigDog developed by Boston Dynamics

Boston Dynamics released video footage of a new generation BigDog as shown in Fig.11 able to walk on icy terrain and recover its balance when kicked from the side.

3. Related work

3.1 Development of the mobile robot system to aid the daily life for physically handicapped (Interface using internet browser)

Yoshiyuki Takahashi and et al. from Shibaura institute of technology developed the mobile robot system (Yoshiyuki Takahashi, et. al., 1998), which could bring daily using objects putting them somewhere in the room with semi-automatically control as shown in Fig.12.

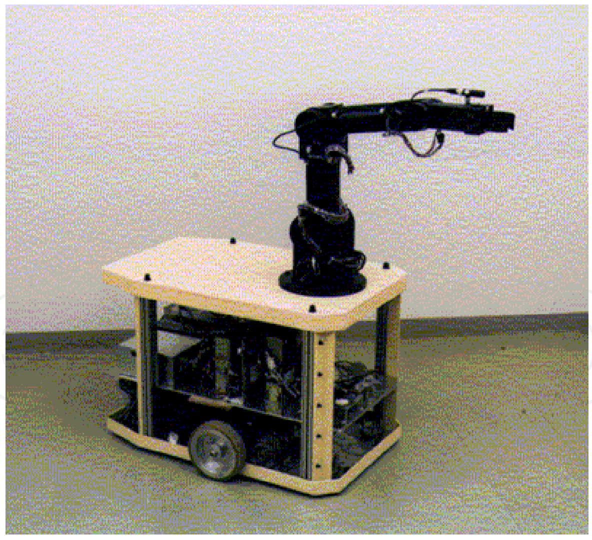


Fig. 12. Mobile robot system developed by Shibaura institute of technology

It is necessary to use graphical and interactive interface to operate the robot, because the operator of this robot is physically handicapped and not always engineer and professional about the robot knowledge as shown in Fig.13.

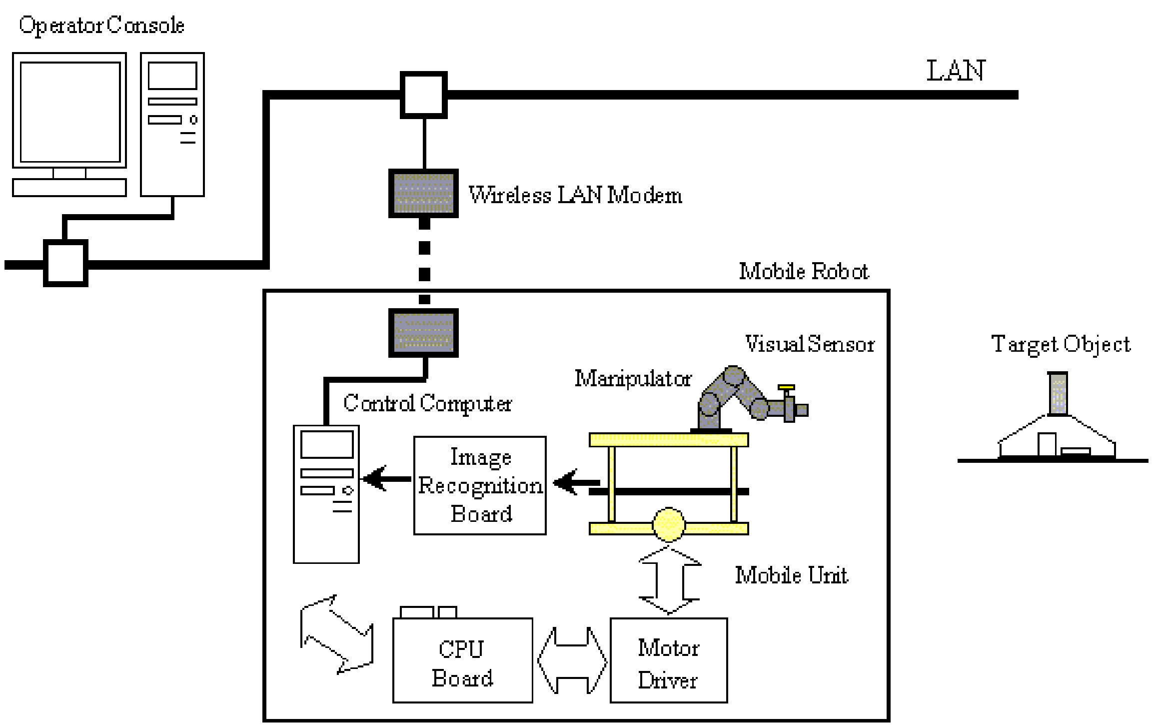


Fig. 13. System structure of mobile robot

3.2 Communication framework for sensor-actuator data in mobile robots

Fernandez proposes the architecture of the robot is designed in Fig.14, which shows several modules inter-connected with CAN bus (Fernandez J., et al., 2007). Each module performs one specific task in the distributed architecture. The actuator and the sensory modules are executed with basic control algorithms.

The communications protocol and CAN master process are implemented in figure 14 showing the different slaves are connected to the master. Control system for all modules worked on PC attached to CAN bus using a CAN-USB adapter.

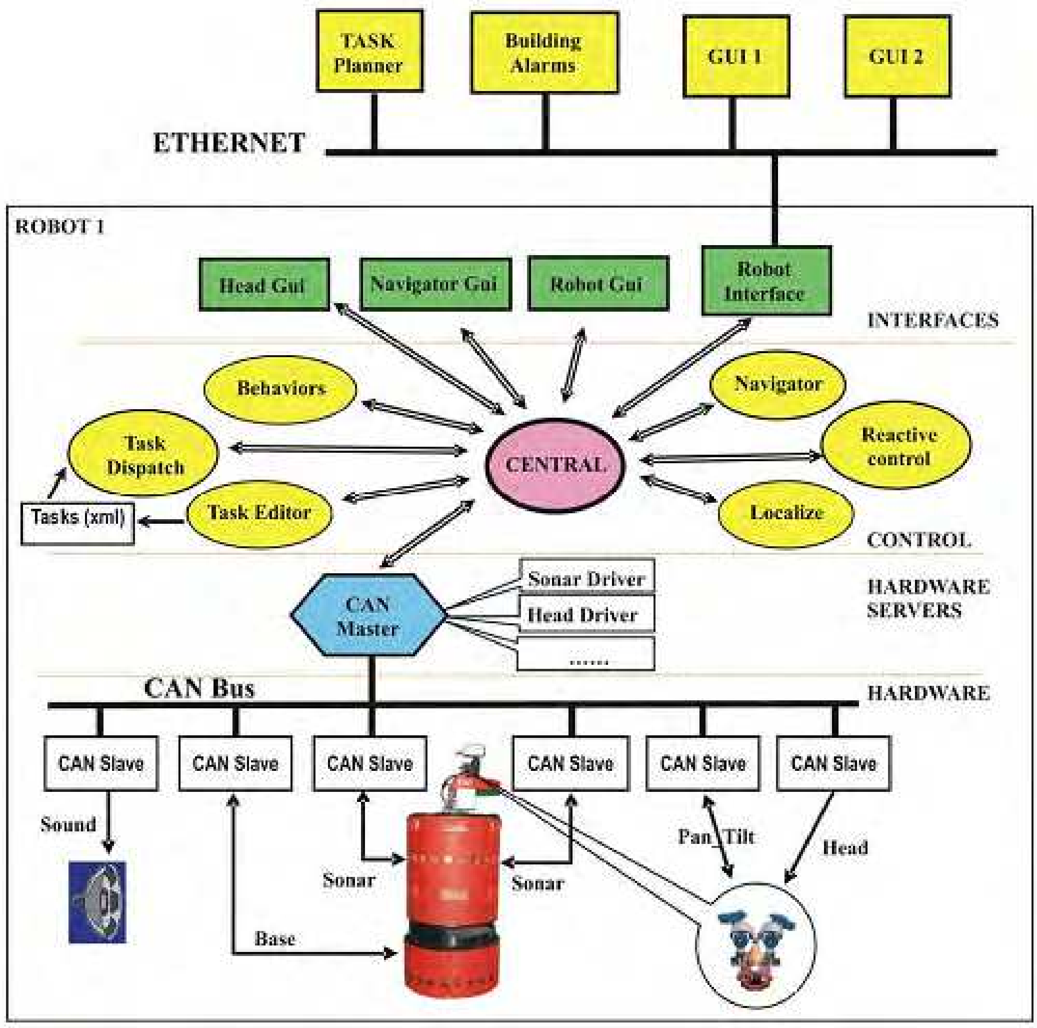


Fig. 14. Robot architecture of CAN system based on sensor and actuator and PC control module communications.

The connection and disconnection of the different slaves by loading the corresponding driver is operated by CAN server. The CAN server will first register and initialize the new module connection, for example if a module with sonar sensors is connected then it will be identified and the module will hand over the messages to the sonar module. In the case of connecting a new module, the master will send information to configure the slave and change the watchdog time. If the master does not receive the watchdog of a slave for a period longer that a timeout, it assumes the slave is disconnected or has some error and it will be notified to the control programs interested in the slave data.

3.3 SMARbot: A miniature mobile robot paradigm for ubiquitous computing

Yan Meng from Stevens Institute of Technology, Hoboken introduce SMARbot paradigm (Meng Yan, et al., 2007). The software reconfiguration of microprocessor and a core component for hardware reconfiguration is implemented in the FPGA. Multiple sensors and actuators with corresponding device drivers and signal processing modules are in the sensor or actuator layer. Each control module consists of one or more input ports, one or more output ports, and any number of other connections. The functionality of the module is implemented to provide automatic integration of the control modules. The information flow, communication and synchronization should be handled automatically by the operating system.

4. The I^{2}C bus system overview

The standard Inter-IC (Integrated Circuit) bus named I^{2}C is shorthand providing a good support for communication with various peripheral devices (Philips Semiconductor, 2000). It is a simple, low-bandwidth, short-distance protocol. There is no need for chip select or arbitration logic, making it cheap and simple to implement in hardware. Most I^{2}C bus devices operate at speeds up to 400 Kbps. The I^{2}C bus system is easy to link multiple devices together since it has a built-in addressing format. The I^{2}C bus is a two wire serial bus as shown in Fig. 15. The two I^{2}C signals are serial data (SDA) and serial clock (SCL).

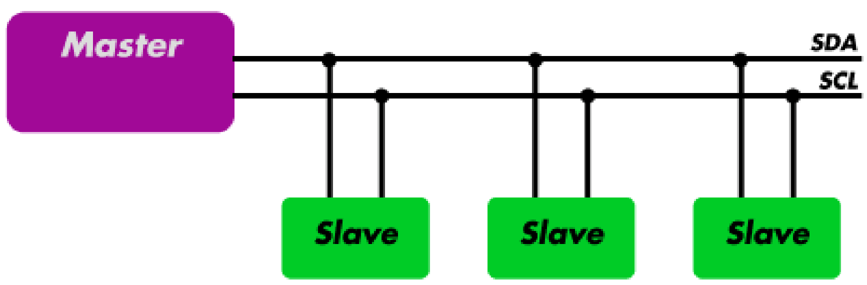


Fig. 15. The I^{2}C bus has only two lines in total

It is possible to support serial transmission of eight-bit bytes with seven-bit bytes device addresses plus control bits over the two wire serial bus. The device called the master starts a transaction on the I^{2}C bus. The master normally controls the clock signal. A device controlled and addressed by the master is called a slave. The I^{2}C bus protocol supports multiple masters, but most system designs include only one. There may be one or more slaves on the bus. Both masters and slaves can receive and transmit data bytes. The slave device with compatible hardware on I^{2}C bus is produced with a predefined device address, which may be configurable at the board device.

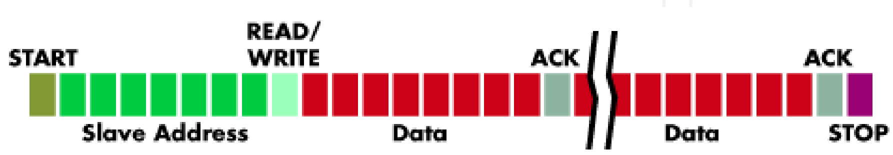


Fig. 16. The I^{2}C bus communication

The master must send the device address of the slave at the beginning of every transaction. Each slave is responsible for monitoring the bus and responding only to its own address. As shown in Fig. 16, the master begins to communicate by issuing the start condition. The master continues by sending seven-bit slave device address with the most significant bit.

The eighth bit (read or write bit) after the start bit specifies whether the slave is now to receive or to transmit information. This is followed by an ACK bit issued by the receiver, acknowledging receipt of the previous byte. Then the transmitter (slave or master, as indicated by the bit) transmits a byte of data starting with the MSB. At the end of the byte, the receiver (whether master or slave) issues a new ACK bit. This 9-bit pattern is repeated if more bytes need to be transmitted. In a write transaction (slave receiving), when the master is done transmitting all of the data bytes it wants to send, it monitors the last ACK and then issues the stop condition. In a read transaction (slave transmitting), the master does not acknowledge the final byte it receives. This tells the slave that its transmission is done. The master then issues the stop condition.

5. Development of mobile robot based on I^2C bus system

In this book chapter, the system of mobile robot named AMRO (Surachai, 2010c) is deeply explained as example for understanding. This robot is developed by student team from Measurement and Mobile Robot laboratory. Its hardware is constructed and combined with the electronic components including the control program.

5.1 Hardware development for AMRO

The mobile robot is designed based on differential drive system (Byoung-Suk Choi 2009; Surachai and et al., 2009) as shown in Fig. 17. The combination of two driven wheels allows the robot to be driven straight, in a curve, or to turn on the spot. The translation between driving commands, for example a curve of a given radius and the corresponding wheel speeds are controlled by software.

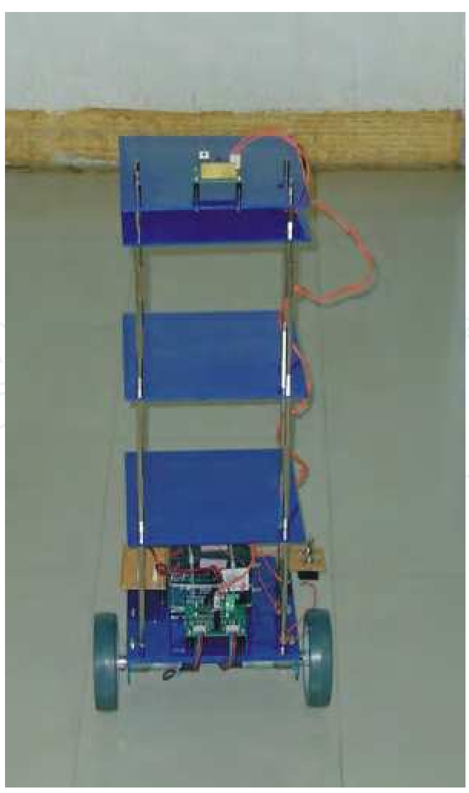


Fig. 17. The Autonomous Mobile Robot (AMRO), Measurement and Mobile robot laboratory

The AMRO is driven by two wheels and a caster powered by an MD25 Dual 5A controller. The MD25 motor driver is designed with 12v battery, which drives two motors with independent or combined control as shown in Fig. 18.

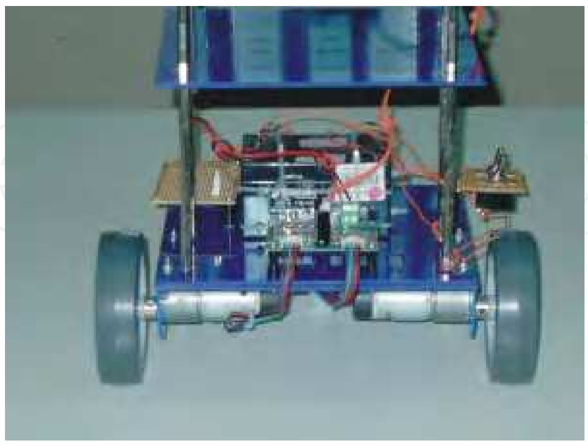


Fig. 18. The MD25 motor driver integrated on AMRO

It reads motors encoders and provides counts for determining distance traveled and direction. Motor current is readable and only 12v is required to power the module. Onboard 5v regulator can supply up to 1A peak, 300 mA continuously to external circuitry Steering feature, motors can be commanded to turn by sent value.

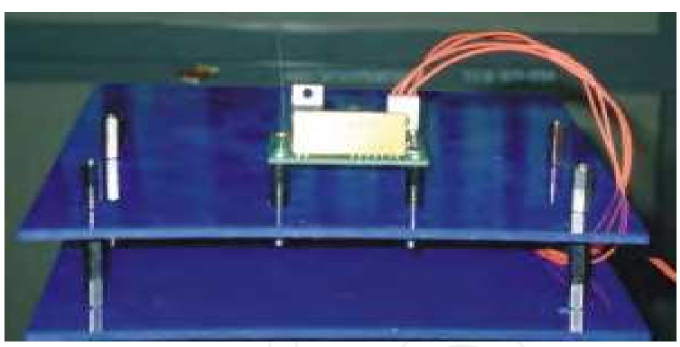


Fig. 19. The CM02 Radio communications module integrated on AMRO

The CM02 Radio communications module as shown in Fig. 19 works together with its companion RF04 module from a complete interface between PC and I^2C devices. The commands can be sent to the robot and receive telemetry data back up to the PC. The CM02 module is powered from battery, which can be anything from 6-12v. There are four I^2C connectors on the CM02, but it is not limited to four I^2C devices. The CM02 radio module provides communication with an RF04 module connected to the PC's USB port. It also provides the MD25 and I^2C devices with 5v supply from its on-board 5v regulator. The AMRO is powered by battery which goes to the CM02 module and also to the MD25 for motor power. All of the modules are connected together with a four wire I^2C loop, which are 5v, 0v, SCL and SDA lines. The PC can now control robot's motors and receive encoder information from AMRO. That means the PC now becomes the robot's brain as shown in Fig.20.

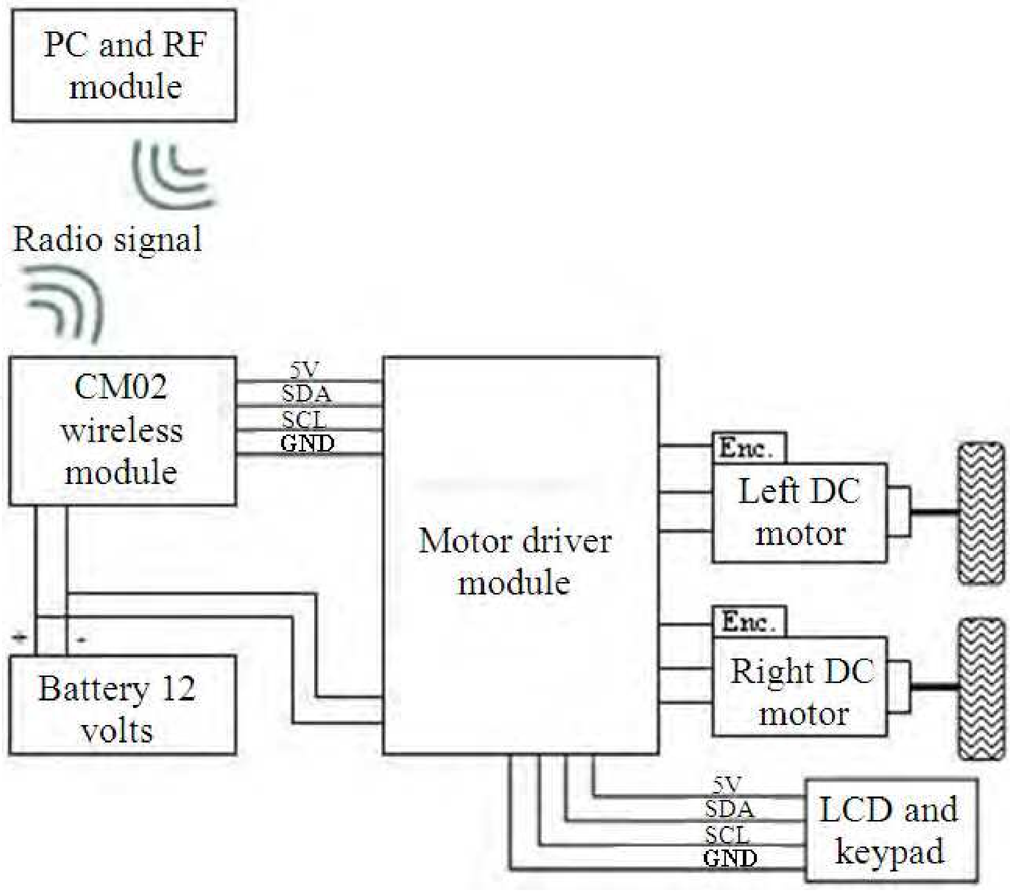


Fig. 20. The AMRO's system control

5.2 The communication between robot and I²C bus devices (Panich, 2008)

Surachai developed the I²C bus system to control information between the robot and sensors or additional devices.

In order that the signal line of serial port from the robot can connect to devices on I²C bus, the electrical master module is designed to generate signal SDA and SCL as shown in Fig. 21.

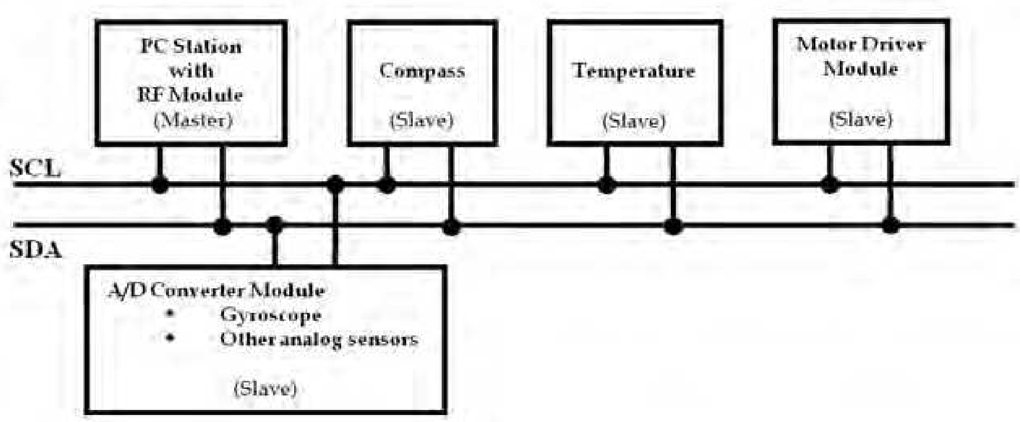


Fig. 21. Hardware communication between Mobile robot (AMRO) and I²C devices

The PC station and RF radio module are selected to produce signal and work as master device. The system of mobile robot can directly connect to I2C devices and other devices, which cannot support I2C system, they can be connected through microcontroller with interface circuit.

5.2.1 The I2C bus devices

Standard I2C devices operate up to 100Kbps, while fast-mode devices operate at up to 400Kbps. A 1998 revision of the I2C specification (v. 2.0) added a high-speed mode running at up to 3.4Mbps. Most of the I2C devices available today support 400Kbps operation. Higher-speed operation may allow I2C to keep up with the rising demand for bandwidth in multimedia and other applications.

5.2.1.1 Compass sensor

The first I2C slave device is compass sensor module as shown in Fig. 22. This sensor can work on the I2C bus without addition circuit. This compass module has been specifically designed for use in robots as an aid to navigation. The aim was to produce a unique number to represent the direction the robot is facing. The compass uses the Philips KMZ51 magnetic field sensor, which is sensitive enough to detect the earth's magnetic field. The output from two of them mounted at right angles to each other is used to compute the direction of the horizontal component of the earth's magnetic field. The compass module requires a 5v power supply at a nominal 15mA. The pulse width varies from 1mS (0°) to 36.99mS (359.9°) – in other words 100uS/° with a +1mS offset. On I2C bus, there is an important consideration that consists of the address from manufacturer and the address from user.

Fig. 22. Compass module slave device

5.2.1.2 Gyroscope sensor

A gyroscope is a device for measuring or maintaining orientation, based on the principles of angular momentum (Komoriya, K. and Oyama, E., 1994). A mechanical gyroscope is essentially a spinning wheel or disk whose axle is free to take any orientation. This orientation changes much less in response to a given external torque than it would without the large angular momentum associated with the gyroscope's high rate of spin. Since external torque is minimized by mounting the device in gimbals, its orientation remains nearly fixed, regardless of any motion of the platform on which it is mounted. Because this gyroscope is not designed for the I2C bus system, it must be connected through microcontroller and read its information as shown in Fig. 23. In order to microcontroller can work on the I²C bus system, it must be specified in I²C format. Now gyroscope with microcontroller works as slave device. This module can read information from gyroscope and send to master device.

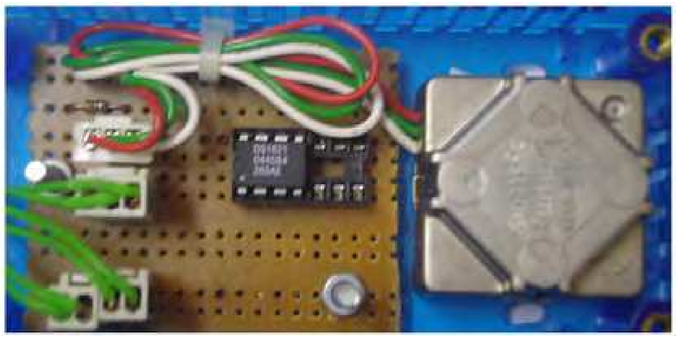


Fig. 23. Gyroscope with microcontroller worked on I²C bus as slave

5.2.1.3 Temperature sensor (DS1621)

The DS1621 as shown in Fig. 24 supports I²C bus and data transmission protocol. A device sends data onto the bus defined as a transmitter, and a device receiving data as a receiver. The device controls the message called a master and devices are controlled by the master called slaves. The bus must be controlled by a master device which generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1621 operates as a slave on the 2-wire bus. Connections to the bus are made via the open-drain I/O lines SDA and SCL. A control byte is the first byte received following the START condition from the master device. The control byte consists of a 4-bit control code set as 1001 binary for read and write operations.

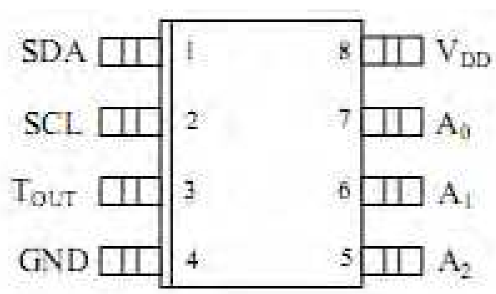


Fig. 24. DS1621 temperature sensor

The next 3 bits of the control byte are the device select bits (A2, A1, A0). They are used by the master device to select which of eight devices are to be accessed. These bits are in effect the 3 least significant bits of the slave address. The last bit of the control byte (R/ W) defines the operation to be performed. When set to a "1" a read operation is selected, when set to a "0" a write operation is selected. Following the START condition the DS1621 monitors the SDA bus checking the device type identifier being transmitted. Upon receiving the 1001 code and appropriate device select bits, the slave device outputs an acknowledge signal on the SDA line.

5.3 Software development for AMRO

A software is developed based on PC application which now manually controls AMRO. The control window contains three sections, Wireless Serial Connection, Manual Control and

Input Velocity. During connection with robot, the PC can send command to the robot and receives sensor information from robot by using I^2C bus system. The START button is used to establish the connection with real robot using wireless serial connection. The STOP button is used to disconnect the robot from the PC. It terminates the communication between robot and PC. The desired velocity can be defined in mm sec^{-1} for forward and backward movement in this Input Velocity block. In this program, the robot velocity are limited up to maximum 500 mm sec^{-1} for safety reason. Left and right turning is done by heading setting, which has already defined in the program. This is the important one to navigate the robot in the environment. As shown in Fig. 25 there are total five buttons to control the robot, which are Forward, Backward, Turn Left, Turn Right and Stop. After establishing connection with the robot through wireless communication and entering all velocity values, the robot will be ready to move in the desired directions. By pressing ***F*** button robot starts to run forward with the given velocity. To stop the robot in forward moving, the ***S*** button must be pressed. The ***B*** button for backward moving works as same as the ***F*** button. The ***L*** and ***R*** button are used to turn the robot in left and right direction respectively. After the ***L*** or ***R*** button are pressed, the ***S*** button can use to stop the robot in desired position. The operation is same like for left button.

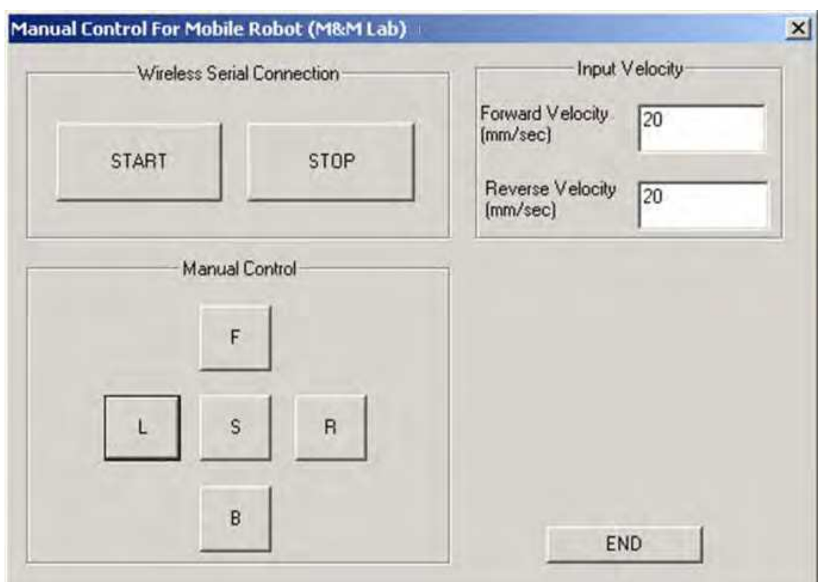


Fig. 25. Manual control window for AMRO

As above mentioned, the ***S*** button is very useful to stop the robot motion. In any condition of the robot, this button plays an important role to restrict the further motion of the robot without disconnecting from PC too. The software is designed to control data between the robot and sensor device, which is programmed based on Visual C^{++}. The software must be able to control SCL and SDA line. To control data line on the I^2C bus, the step ordering of function based on Visual C^{++} must be carefully accurately considered and programmed, because if one step miss or does not complete, all devices on this bus will fail. The main function used for programming will be now detailed. All conditions are only generated by robot (as master device). The two main functions are I2C_START ( ) and I2C_STOP ( ). The I2C_START ( ) function produces the START condition as shown in Fig.26.

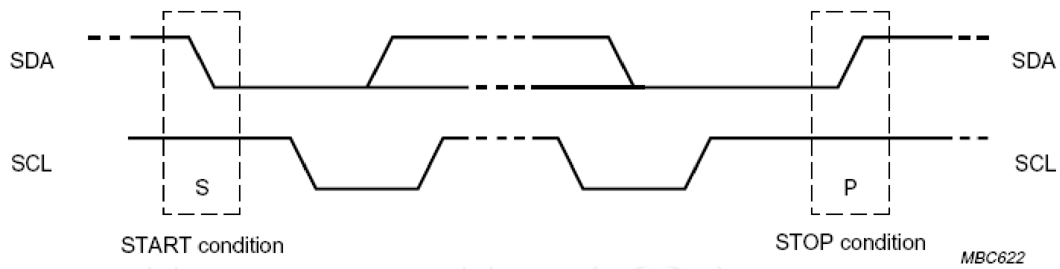


Fig. 26. Start and stop condition of two lines

This condition is a HIGH to LOW transition on the SDA line while SCL is HIGH. And the I2C_STOP ( ) will produce the STOP condition. This procedure is a LOW to HIGH transition on the SDA line while SCL is HIGH. Before the START condition will begin and after STOP condition finished, the bus is considered to be always free condition, the both lines must be HIGH. The next main function is I2C_ACK ( ). As shown in Fig.27, after the robot (master device) sent data to slave device finished, the slave device must send back the acknowledgement that the slave device received data already.

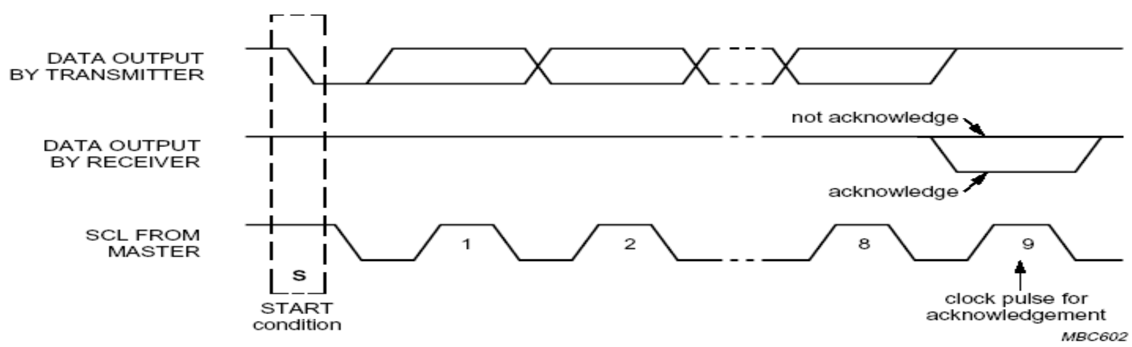


Fig. 27. Acknowledge condition

As shown in Fig.28, it shows a complete transfer cycle associated with a frame of data. Firstly, the master initiates a write by asserting logic-0 at bit-8, where a slave address is defined by the other 7-bits. A acknowledge signal then follows from the slave as specified in bit-9. The second and third bytes are the data and acknowledge signal. The 7-bits addressing allows 127 devices on the I²C bus, by using 2-bytes address, which can be extended further. The last two main functions are to control and get data from slave device. It consists of I2C_SEND ( ) and I2C_RECEIVE ( ) function. This I2C_SEND ( ) function is sent always by robot (master device) to control and set slave devices configuration and the I2C_RECEIVE ( ) function is also sent by robot to receive data from slave devices.

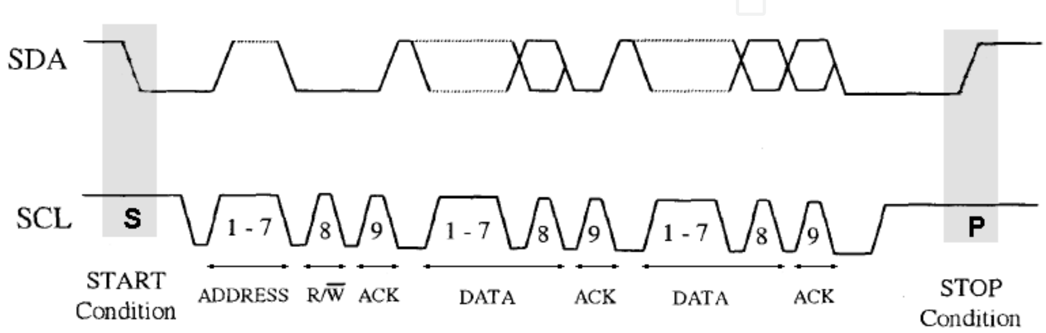


Fig. 28. Data transfer on the I²C bus

The examples of function ordering to connect a slave device through I²C bus to receive data are detailed below.

5.3.1 Function ordering for encoder information reading

The encoders are mounted at motor and their information is read by motor controller (MD25). The MD25 is designed to operate in a standard I²C bus system. To read encoder information, the function ordering can follow as below.

Step 1 : I2C_START ( ),
Step 2 : I2C_SEND ( ) – Call encoder address and write mode configuration,
Step 3 : I2C_ACK ( ),
Step 4 : I2C_SEND ( ) – Write the register number,
Step 5 : I2C_ACK ( ),
Step 6 : I2C_START ( ),
Step 7 : I2C_SEND ( ) – Call encoder address again,
Step 8 : I2C_ACK ( ),
Step 9 : I2C_RECEIVE ( ),
Step 10 : I2C_ACK ( ),
Step 11 : I2C_STOP ( ).

By using the feedback information from the two encoders on the left and right wheels of mobile robot, the position and heading angles of the mobile robot can be estimated. The distance and heading increment can be obtained as follows. (Hye Ri Park, et. Al., 2009; Surachai Panich, 2010a; Surachai Panich, 2010b; Surachai Panich and Nitin V. Afzulpurkar, 2011).

$$\Delta d_k^{encoder} = \frac{\Delta d_{R,k}^{encoder} + \Delta d_{L,k}^{encoder}}{2} \tag{1}$$

$$\Delta \psi_k^{encoder} = \frac{\Delta d_{R,k}^{encoder} - \Delta d_{L,k}^{encoder}}{B_k} \tag{2}$$

Then the position can be estimated as

$$X_{k+1} = X_k + \Delta d_{x,k}^{encoder} \tag{3}$$

$$Y_{k+1} = Y_k + \Delta d_{y,k}^{encoder} \tag{4}$$

$$\psi_{k+1} = \psi_k + \Delta \psi_k^{encoder} \tag{5}$$

where

$$\Delta d_{x,k}^{encoder} = \Delta d_k^{encoder} * \cos\psi_k$$

$$\Delta d_{y,k}^{encoder} = \Delta d_k^{encoder} * \sin\psi_k$$

The estimated position and heading can be detailed in software called PAMRO developed by Visual C++ as shown in Fig. 28.

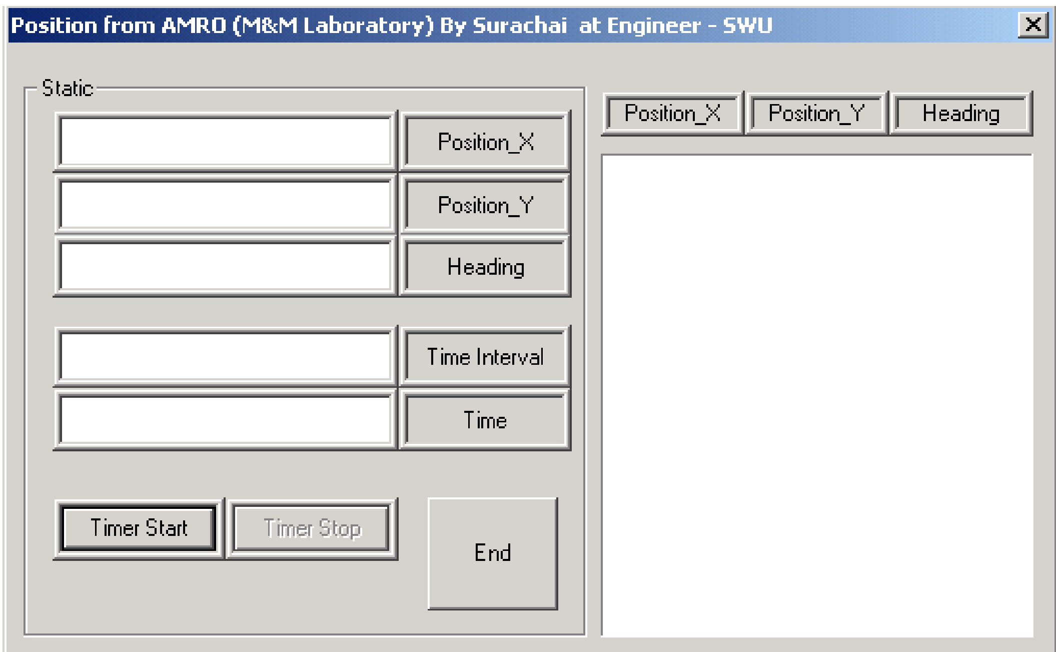


Fig. 28. The software PAMRO

5.3.2 Function ordering for compass information reading

The function ordering for compass module is detailed below.

Step 1 : I2C_START ( ),
Step 2 : I2C_SEND ( ) – Call compass address from robot and write mode configuration,
Step 3 : I2C_ACK ( ),
Step 4 : I2C_SEND ( ) – Write the register number,
Step 5 : I2C_ACK ( ),
Step 6 : I2C_START ( ),
Step 7 : I2C_SEND ( ) – Call compass address again,
Step 8 : I2C_ACK ( ),
Step 9 : I2C_RECEIVE ( ),
Step 10 : I2C_ACK ( ),
Step 11 : I2C_STOP ( ).

5.3.3 Function ordering for gyroscope information reading

To read gyroscope information, it needs device to convert analog to digital signal, because this gyroscope is not designed to work on I²C bus system. The IC-PCF8591is selected supported the I²C bus system to convert analog signal from gyroscope information.

Step 1 : I2C_START ( ),
Step 2 : I2C_SEND ( ) – Call slave device address from robot and write mode configuration,

Step 3 : I2C_ACK ( ),
Step 4 : I2C_SEND ( ) – Enable analog output, single mode and analog channel selection,
Step 5 : I2C_ACK ( ),
Step 6 : I2C_STOP ( ),
Step 7 : I2C_START ( ),
Step 8 : I2C_SEND ( ) – Read mode; start to read data from analog input of selected channel,
Step 9 : I2C_ACK ( ),
Step 10 : I2C_RECEIVE ( ),
Step 11 : I2C_ACK ( ),
Step 12 : I2C_STOP ( ).

5.3.4 Function ordering for temperature reading

To read temperature, it must be programmed in 3 steps; start to convert data, stop to convert data and read data.

Start to convert data:

Step 1 : I2C_START ( ),
Step 2 : I2C_SEND ( ) – Call slave device address from robot and write mode configuration,
Step 3 : I2C_ACK ( ),
Step 4 : I2C_SEND ( ) – Send command register, start to convert data,
Step 5 : I2C_ACK ( ),
Step 6 : I2C_STOP ( ),

Stop to convert data:

Step 1 : I2C_START ( ),
Step 2 : I2C_SEND ( ) – Call slave device address from robot and write mode configuration,
Step 3 : I2C_ACK ( ),
Step 4 : I2C_SEND ( ) – Send command register, stop to convert data,
Step 5 : I2C_ACK ( ),
Step 6 : I2C_STOP ( ),

Read data:

Step 1 : I2C_START ( ),
Step 2 : I2C_SEND ( ) – Call slave device address from robot and write mode configuration,
Step 3 : I2C_ACK ( ),
Step 4 : I2C_SEND ( ) – Send command register, stop to convert data,
Step 5 : I2C_ACK ( ),
Step 6 : I2C_START ( ),
Step 7 : I2C_SEND ( ) – Send command register, stop to convert data,
Step 8 : I2C_ACK ( ),
Step 9 : I2C_RECEIVE ( ) – Read MSB of temperature register,

Step 10 : I2C_ACK ( ),
Step 11 : I2C_RECEIVE ( ) – Read LSB of temperature register,
Step 12 : I2C_ACK ( ),
Step 13 : I2C_STOP ( ),

With these steps, the robot can get data from only one module for one time. If the robot wants to get data from this module once again or from another module, the robot must rerun once again from first to last step. The software is developed by Visual C^{++} to read information from I^2C devices as shown in Fig.29.

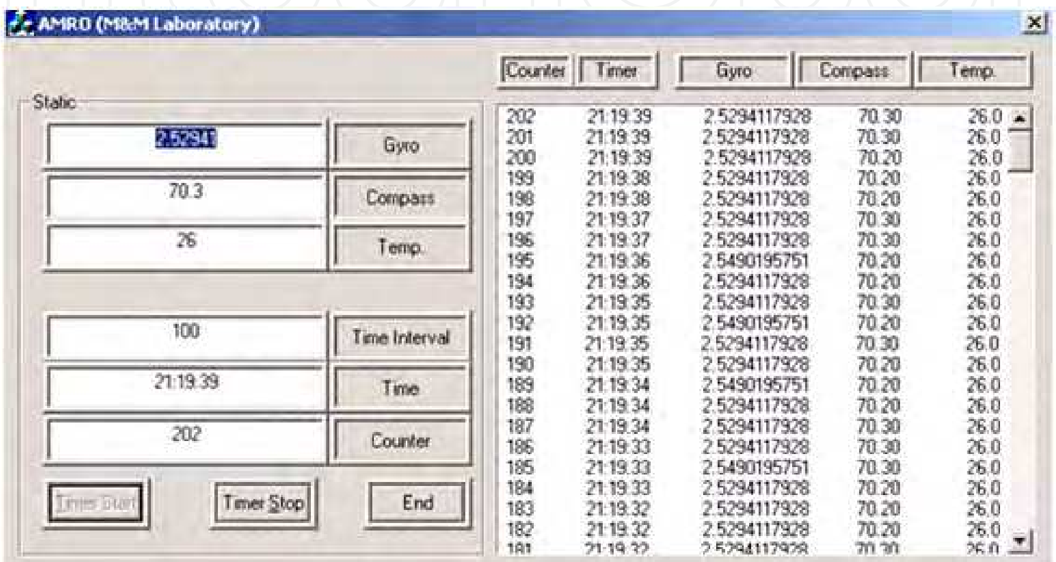


Fig. 29. Software to read data from sensors integrated on AMRO by I^2C bus system

6. Conclusion

For this book chapter has mainly purpose to introduce basic structure of mobile robot and communication between mobile robot and sensors by using I^2C bus system. The mobile robot named AMRO is introduced as example. Its hardware is constructed and combined with the electronics components. The real robot is tested successfully using manual controls (GUI) developed by using C^{++} programming. This small GUI is very useful for new user to test the various robot movements. The software is developed with manual controls for real robot. After establishing connection with the robot through wireless communication and entering all velocity values, the robot will be ready to move in the desired directions. By pressing ***F*** button robot starts to run forward with the given velocity. The I^2C bus system is selected for main system of our mobile robot, because it can be conveniently developed and modified, if new sensors must be integrated or more analog sensors are used later. The I^2C bus system consists of master and slave devices that the master device in this work is PC station and ADC-Converter for analog sensors (gyroscope), compass module and temperature sensor work as slave devices integrated on the mobile robot (AMRO). The I^2C bus can work very well and has no problem with AMRO system. The software is developed to control the robot and to read all analog sensors based on the I^2C bus format. The developed software is programmed based on Visual C^{++}. The software devleopment must be programmed in the I^2C bus format to control SDA and SCL lines.

7. Acknowledgement

This research from Measurement and Mobile Robot Laboratory (M & M LAB) was supported by Faculty of Engineering, Srinakharinwirot University under grant 180/2552.

8. References

Byoung-Suk Choi (2009). Mobile Robot Localization in Indoor Environment using RFID and Sonar Fusion System, *IEEE/RSJ International Conference on Intelligent Robots and Systems,* October 11-15, 2009 St. Louis, USA

Fernandez J., et al., 2007. Communication framework for sensor-actuator data in mobile robots, *ISIE 2007. IEEE International Symposium on Industrial Electronics, 2007*, Digital Object Identifier: 10.1109/ISIE.2007.4374825 Publication Year: 2007 , Page(s): 1502 – 1507

Hye Ri Park, et al., 2009. A Dead Reckoning Sensor System and a Tracking Algorithm for Mobile Robot. *ICROS-SICE International Joint Conference 2009*, August 18-21, Fukuoka International Congress Center, Japan.

Komoriya, K. and Oyama, E., 1994. Position estimation of a mobile robot using optical fiber gyroscope (OFG). *Proceedings of the Intelligent Robots and Systems, Advanced Robotic Systems and the Real World, IROS '94.* IEEE/RSJ/GI International Conference on 12-16 Sept. 1994.

Meng Yan, et al., 2007. SMARbot:_A_Miniature_Mobile_Robot_Paradigm_for_Ubiquitous Computing, *The 2007 International Conference on Intelligent Pervasive Computing, 2007 IPC*, Digital Object Identifier: 10.1109/IPC.2007.80 Publication Year: 2007, Page(s): 136 – 139

Panich, S. (2008). A mobile robot with an inter-integrated circuit system, *10th International Conference on Control, Automation, Robotics and Vision,* Publication Year: 2008, Page(s): 2010 – 2014, Digital Object Identifier: 10.1109/ICARCV.2008.4795839

Philips Semiconductor (2000), *I^2C Bus Specification*. Version 2.1, 2000.

Surachai, P.; Thiraporn, T.; Chaiyaporn, L.; Kittichai, T.; Tayawat, S., 2009. Sensor fusion for differential encoder integrated with light intensity sensors and accelerometer. *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA),* DOI: 10.1109/CIRA.2009.5423180, Page(s): 349 – 354

Surachai Panich (2010a). Mobile Robot Driven by Odometry System Integrated with Accelerometer. *Far East Journal of Electronics and Communications*, Volume 4, Issue 2 (June 2010) Page: 113 – 122

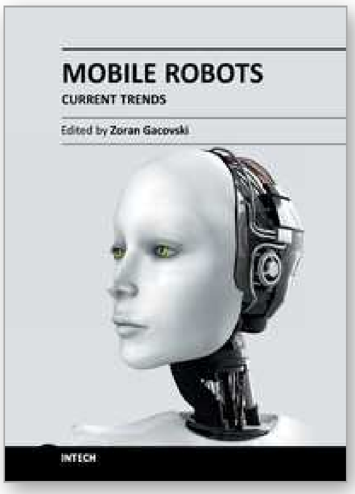
Surachai Panich (2010b). Dynamics, Control and Simulation of Robots with Differential Drive Integrated with Light Intensity Sensors. *Far East Journal of Dynamical Systems*, Volume 13, Issue 2 (July 2010) Page: 157 – 164

Surachai Panich (2010c). Mathematic model for mobile robot. *Far East Journal of Mathematical Sciences (FJMS),* Volume 46, Issue 1 (November 2010) Page: 23 – 32

Surachai Panich and Nitin V. Afzulpurkar (2011), Sensor Fusion Techniques in Navigation Application for Mobile Robot, in *Sensor Fusion - Foundation and Applications*, edited by Dr. Ciza Thomas, pp. 101-120, Published by InTech - Open Access Publisher, ISBN 978-953-307-446-7, Rijeka, Croatia.

Wikipedia (June 2011). Mobile robot, *Wikipedia; the free encyclopedia*, Available URL: http://en.wikipedia.org/wiki/Mobile_robot

Yoshiyuki Takahashi, et al., 1998. Development of the Mobile Robot System to aid the daily life for physically handicapped (Interface using internet browser), *The TIDE 98 Congress, Technology for Inclusive Design and Equality*, July, Helsinki



Mobile Robots - Current Trends

Edited by Dr. Zoran Gacovski

ISBN 978-953-307-716-1
Hard cover, 402 pages
Publisher InTech
Published online 26, October, 2011
Published in print edition October, 2011

This book consists of 18 chapters divided in four sections: Robots for Educational Purposes, Health-Care and Medical Robots, Hardware - State of the Art, and Localization and Navigation. In the first section, there are four chapters covering autonomous mobile robot Emmy III, KCLBOT - mobile nonholonomic robot, and general overview of educational mobile robots. In the second section, the following themes are covered: walking support robots, control system for wheelchairs, leg-wheel mechanism as a mobile platform, micro mobile robot for abdominal use, and the influence of the robot size in the psychological treatment. In the third section, there are chapters about I2C bus system, vertical displacement service robots, quadruped robots - kinematics and dynamics model and Epi.q (hybrid) robots. Finally, in the last section, the following topics are covered: skid-steered vehicles, robotic exploration (new place recognition), omnidirectional mobile robots, ball-wheel mobile robots, and planetary wheeled mobile robots.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Surachai Panich (2011). Development of Mobile Robot Based on I2C Bus System, Mobile Robots - Current Trends, Dr. Zoran Gacovski (Ed.), ISBN: 978-953-307-716-1, InTech, Available from: http://www.intechopen.com/books/mobile-robots-current-trends/development-of-mobile-robot-based-on-i2c-bus-system

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821