

Intelligent wheeled mobile robots for blind navigation application

214

Received 30 August 2015
Revised 18 January 2016
Accepted 5 September 2016

Ter-Feng Wu

Department of Electrical Engineering, National Ilan University, Yilan, Taiwan

Pu-Sheng Tsai

*Department of Electronic Engineering,
China University of Science and Technology, Taipei, Taiwan*

Nien-Tsu Hu

*Department of Chemical Systems Research Division,
National Chung-Shan Institute of Science and Technology, Taoyuan, Taiwan, and*

Jen-Yang Chen

Department of Electronic Engineering, Ming Chuan University, Taoyuan, Taiwan

Abstract

Purpose – Visually impaired people have long been living in the dark. They cannot realize the colorful world with their vision, so they rely on hearing, touch and smell to feel the space they live in. Lacking image information, they face challenges in the external environment and barrier spaces. They face danger that is hundreds of times higher than that faced by normal people. Especially during outdoor activities, they can only explore the surrounding environment aided by their hearing and crutches and then based on a vague impression speculate where they are located. To let the blind confidently take each step, this paper proposes sticking the electronic tag of the radio-frequency identification (RFID) system on the back of guide bricks.

Design/methodology/approach – Thus, the RFID reader, ultrasonic sensor and voice chip on a wheeled mobile robot link the front end to the crutch. Once the blind person nears a guide brick, the RFID will read the message on the tag through the voice broadcast system, and a voice will inform the visually impaired person of the direction to walk and information of the surrounding environment. In addition, the CMOS image sensor set up in the wheeled mobile robot is used to detect the black marking on the guide brick and to guide the blind to walk forward or turn around between the two markings. Finally, the lithium battery charging control unit was installed on the wheeled mobile robot. The ATtiny25 microcontroller conducts the battery charge and discharge control and monitoring of the current battery capacity.

Findings – The development of this system will let visually impaired people acquire environmental information, road guidance function and nearby traffic information.

Originality/value – Through rich spatial environment messages, the blind can have the confidence and courage to go outside.

Keywords CMOS image sensor, Embedding microcontroller unit (EMCU), Radio-frequency identification module, Ultrasonic sensor, Wheeled mobile robot

Paper type Research paper



1. Introduction

The most common mobility aid for guiding visually impaired persons is the so-called “white cane”. Its meaning represents respect for the visually impaired, pedestrians on the road must

take the initiative to assist and give way to the blind and all types of vehicles must give way or slow down when they come across the “white cane”. The visually impaired people can detect obstacles on the ground, uneven pavement, potholes, stairs, etc., in the state of the environment through the use of a cane, but this ability requires extensive training and learning. The white cane can only detect a meter ahead at a range of about 45 degrees or so to the left and right and is unable to detect any obstacles above the knee. As uneven pavement and randomly parked vehicles at the roadside can be found everywhere in the city, using the white cane can be really dangerous and difficult. Guide dogs, in addition to being the eyes for the visually impaired people, also play the role of affectionate and inseparable companions.

To help the blind, the design of many electronic aids to guide the blind has also gotten good results. In foreign countries, such devices include the NavChair (Levine *et al.*, 1999) of the Robotics Laboratory of the University of Michigan, the NavBelt (Shoval *et al.*, 1998, 2003) published by Shoval *et al.* (1998) and various types of guide canes (Borenstein, 2001; Gupta *et al.*, 2015). For the blind, the guide brick is an important guiding device. The blind may touch these guide bricks containing much information to achieve the purpose of guiding actions. The guide bricks are divided into the “linear bricks” for guiding and the “point-like brick” for warning and cautionary use. In fact, the messages that the current guide brick can provide are limited to guiding and warning. The help for the blind can be described as inadequate. Moreover, the guide brick settings in many sidewalks had become a row of neatly arranged motor cycles, and the guide bricks have long been damaged and unusable. The biggest difficulty of visually impaired people lies in the identification of direction and the acquisition of information. As long as they are given the proper information and orientation training, visually impaired people can usually move freely. For this reason, the blind guide robots in this plan will combine the radio-frequency technology (radio-frequency identification [RFID]) and voice broadcast system to replace the traditional guide brick facility, thereby providing richer environmental information and plane orientation for the blind and even a guiding map voice in some occasions. In recent years, RFID technology has had substantial progress. RFID card tags store messages in the guide brick, and the voice broadcast system provides the blind orientation identification, bus information, building names or obstacle warnings.

The RFID system (Sarma *et al.*, 2003; Polycarpou *et al.*, 2012) is a non-contact identification technology, which uses radio frequency to identify the card tag on the target object to conduct identification and the work of capturing relevant information. RFID was developed and used by the British in the Second World War (1948) to identify the enemy fighters on the air field. A maturing technology makes the application level increasingly widespread and gradually changes people's consumption patterns and lifestyles. Examples include sensor cards of community access control systems, library bookplates, identification chips implanted in pets and so on. The non-contact manner of use and the successful solving of the power problem are all major technological breakthroughs. Moreover, this revolutionary technology is considered a major breakthrough in the inventory management of logistics retail business. In 2004, the global retail leader Wal-Mart Department Stores began requiring 100 of its largest suppliers to introduce RFID systems in shipment products, leading to the logistics revolution. In the future, if Wal-Mart fully introduces RFID, it is estimated that the entire supply chain can save more than US\$800m, and it is precisely because RFID can create such a high effectiveness that many manufacturers, such as IBM, HP and so on, are actively involved in the development of RFID.

This paper attempts to use the small wheeled mobile robot as a vehicle equipped with CMOS image sensor, ultrasonic sensor, RFID system, voice broadcast and other components

for the experiment. The goal was to replace the white cane and human visual functions with a robot that can smoothly guide a person to the correct path. The embedded microcontroller unit ATmega162 is the core chip of the whole system. With a small size and low power consumption, the chip has challenges with the CMOS image processing algorithms, identification of the two white markings on the guide bricks, RFID, ultrasound detection and voice broadcasts and use of an internal pulse width modulation (PWM) signal generator to send pulse width signals to drive the left and right servo motors and to complete the right walking path of the wheeled mobile robot.

2. Blind guide robot system structure

The overall structure of the blind guide robots can be divided into three parts: the main system control unit, the image processing control unit and the battery charging control unit.

The main results can be summarized as follows:

- The main system control unit contains the embedded microcontroller unit ATmega162 to conduct signal detection processing, respectively, for ultrasonic sensors and the RFID system, and controls the voice broadcast system by synchronous serial communication (Serial Peripheral Interface [SPI]).
- The second part of the image processing control unit has a CMOS image sensor, drawing-mode LCD modules, external expansion memory and so on. The second embedded microcontroller unit ATmega162 is used as the control core of this unit. The CMOS image sensor captures the two black markings on the guide brick, and after the data fitting algorithm processing is completed, the black markings and characteristic parameters are displayed on the drawing-type LCD, and RS232 transmits the control commands to the main system control unit, which controls the wheeled mobile robot's forward, backward, leftward and other related actions.
- The battery charging control unit contains a lithium battery itself and lithium battery charger. The charger itself uses the Buck converter circuit's (switching circuit) charging principles, Smart Battery's charging concept, lithium battery's self-protection circuit and the temperature detector's streamlined circuit ([Figures 1](#) and [2](#)).

2.1 The main system control unit

2.1.1 Left and right servo motor control mode. The Parallax Standard Servo used in this system is a servo motor designed by Parallax Company. It has the advantages of being small in size and light-weight, produces a large torque, uses simple PWM signals as the control and is power-saving. The servo motor uses the control signals of cycle 20-ms PWM. The pulse width is as shown in [Figure 3](#).

For the servo motor wiring, refer to [Figure 4](#). INPUT pin is connected to the ATmega162 microcontroller's PWM output control terminal. The internal PWM function of ATmega162 is used to generate a pulse modulation signal to control the rotation servo motor's forward (clockwise) rotation, reverse (counterclockwise) rotation, stop and other actions.

The embedded microcontroller unit ATMega162 used in the system has two 16-bit timers/counters (Timer1, Timer3), and an 11.05962-MHz quartz oscillator was selected as the internal clock reference. Using Timer3 can generate two sets of PWM drive signals, in which PD4 (OC3A) is used to drive the right wheel servo, and PB4 (OC3B) is for driving the left wheel servo. As to the embedded microcontroller unit ATmega162, ICR3H and ICR3L

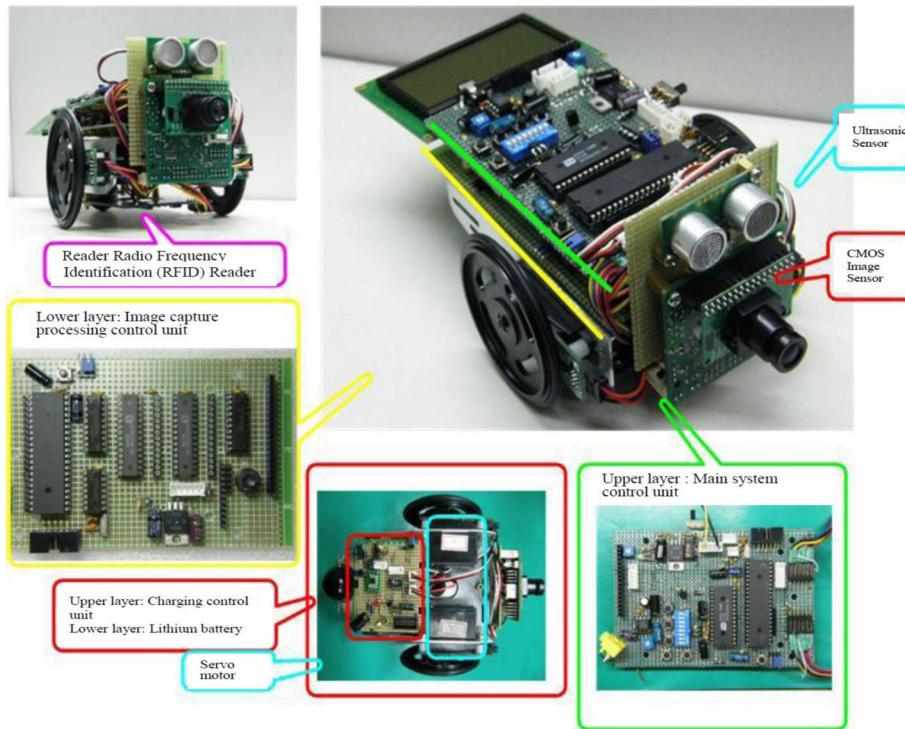


Figure 1.
Blind guide robot
system configuration
diagram

registers control the PWM pulse signal cycle, and OCR3AH, OCR3AL, OCR3BH and OCR3BL registers control the high-state pulse output of the two sets of PWM signals, which is calculated as follows (Atmel, 2013a, 2013b):

$$\text{PWM pulse cycle} = (ICR3H, ICR3L) \times \frac{1}{f_{osc}} \times 2 \times \text{Pre_scale} \times N \quad (1)$$

$$\text{PWM high-state pulse width} = (OCR3AH, OCR3AL) \times \frac{1}{f_{osc}} \times 2 \times \text{Pre_scale} \times N \quad (2)$$

Wherein $f_{osc} = 11.05962\text{MHz}$ is the operating frequency of the external oscillator. Pre_scale is Timer3 as the frequency division coefficient. Herewith it makes Pre_scale = 2. N is the system's frequency division coefficient, N = 1.

$(ICR3H, ICR3L)$ = Input Capture Register 3 (ATmega162 internal register)

$(OCR3AH, OCR3AL)$ = Output Compare Register 3 (ATmega162 internal register)

In addition, we hope to be able to get the relationship between the PWM pulse width and servo rotation speed. The ATMega162 embedded microcontroller unit's internal

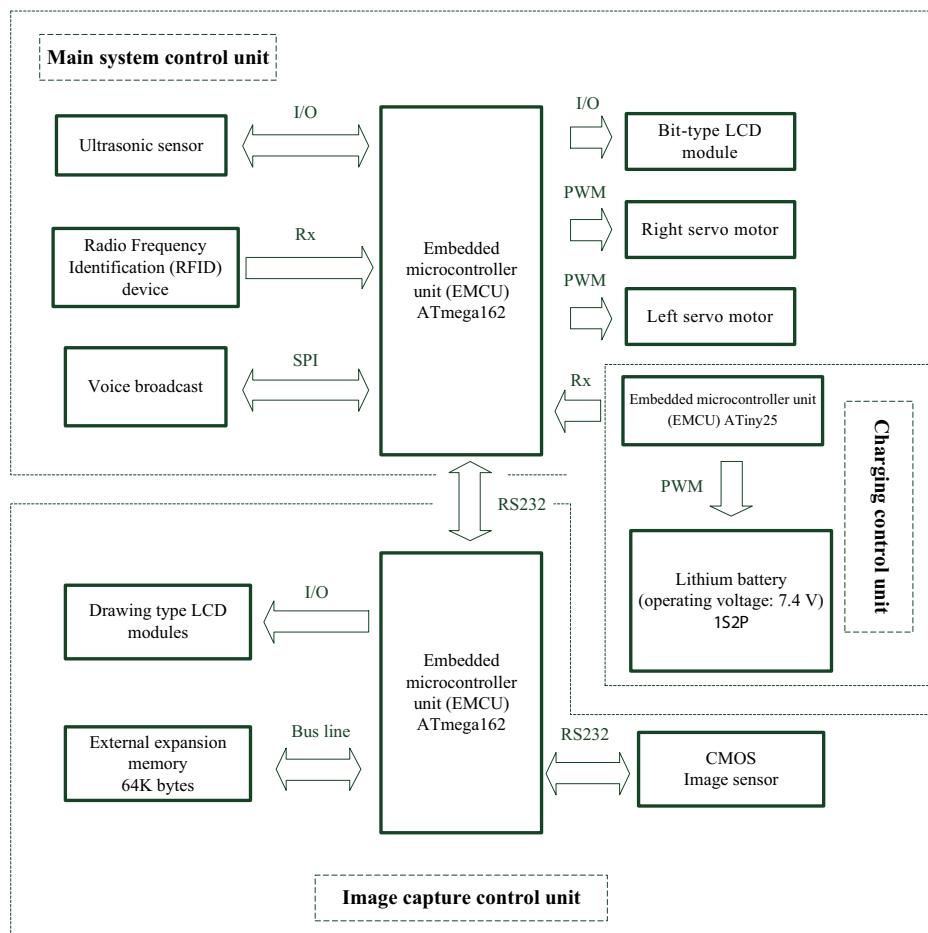


Figure 2.
System structure
block diagram

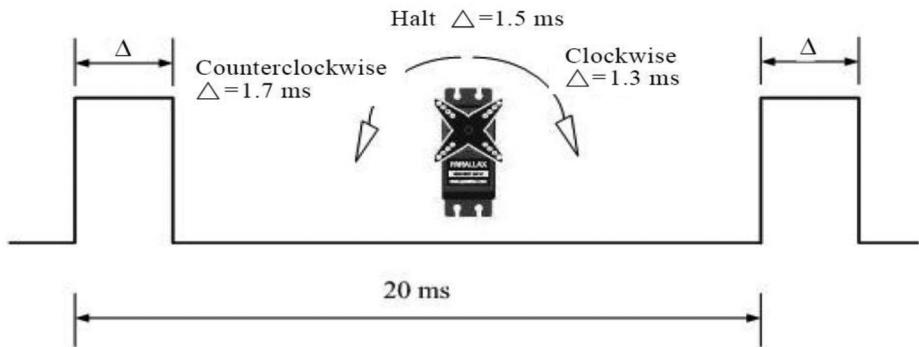
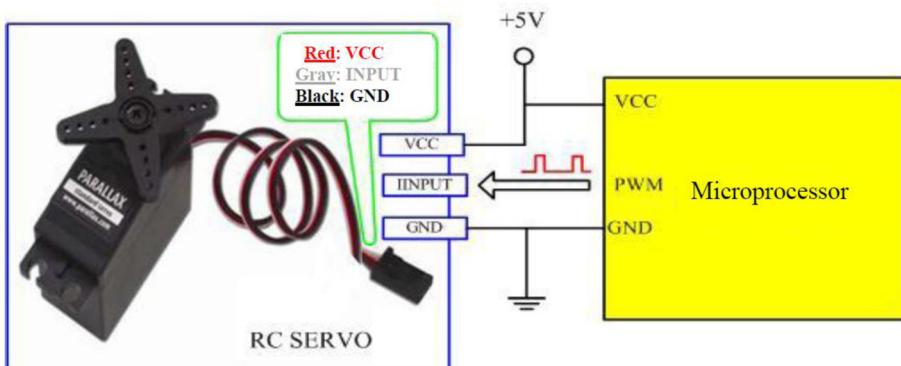
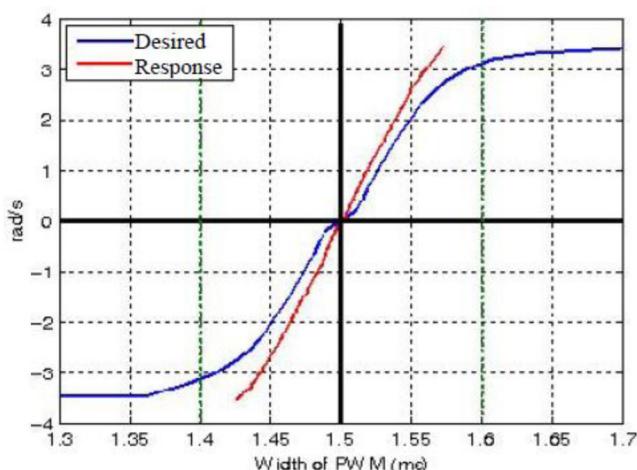


Figure 3.
PWM control signal


Figure 4.
Wiring diagram of
the servo motor

PWM and counter function were used to complete the experiment. The startup PWM generator via OC3A and OC3B pins sends adjustable signals with a cycle of 20 ms and pulse width of $1.3 \sim 1.7$ ms to, respectively, drive the left and right wheel servos and drive the wheel rotation. Enabling the external counter, T0 and T1 at intervals of Δt seconds read the number $n_x(x = 1, 2)$ of pulses transmitted by the optical encoders of the left and right wheels. Assuming that for each revolution of the wheel, the optical encoder will send out N pulses, then the servo motor speed is $(2\pi n_x)/(\Delta t \cdot N)$ (rad/s). For the relevant principle of the optical encoder, please refer to Noetic Design (2005). As shown in Figure 5, when the PWM pulse width is between 1.6 and 1.4 ms, it is the best interval for the servo motor speed and PWM pulse width to maintain a linear relationship. The maximum speed within this range can reach up to around 4.2 rad/s. These data in the following simulation experiment will be taken as the constraint condition of the maximum speed of our wheel.


Figure 5.
The characteristics
curve diagram of
servo rotation speed
and pulse width of
PWM

2.1.2 Ultrasonic sensor module. The ultrasonic sensor SRF05 module used in this system was produced by Devantech Company (Devantech, 2008). It has the following advantages:

- *Ultra-small*: The size is only $43 \times 20 \times 17$ mm (height).
- *Ultra-low operating current*: Under normal DC + 5 V operating voltage, power consumption is 4 mA.
- *Small measurement error*: Kept within 3 per cent.
- *Large measurement range*: 1 cm ~ 4 m range can be measured.
- *Simple operation*: Using the responded pulse width signal can convert the measured distance.

The system uses the operating process of Mode 1. The relevant control timing is as shown in [Figure 6](#). The ultrasonic sensor's input trigger pulse signal (Pin3: Trigger in) and output pulse width signal (Pin2: Echo out) were connected to the ATmega162 microcontroller's Trigger I/O pin and the input capture control pin (ICP). [Figure 6](#) shows the ultrasonic sensor's pin control lines. First, the ATmega162 transmits a high-state pulse signal of $10\ \mu s$ to the trigger pulse input (Trigger in) pin of the ultrasonic module. The interior will generate eight signals of 40-KHz cycles and convert the output pulse width signal (Echo out) pin's low potential to a high potential. When the ultrasonic signals hit the object and rebound to the sensor module, the pulse width signal pin will echo out and revert to a low potential. At this point, we can use the output pulse width to calculate the distance between ultrasonic sensors and the obstacle object. As shown in the specifications provided by the manufacturer, the distance (cm) between ultrasonic sensors and obstacle objects is the response of the high-state pulse signal width multiplied by a scaling factor of $1/58(cm/\mu s)$. As the response pulse signal's width time μs is about $100\ \mu s \sim 25ms$, from these data, the measurement distance range of about $1.72 \sim 431$ cm can be obtained.

2.1.3 Radio-frequency identification system. A complete RFID system is composed of three parts: a reader; an electronic tag, also called the transponder; and a computer application database. The principle of operation is that the reader transmits specific RF signals and wirelessly receives tag information, and at the same time, the RF signals become radio wave energy that drives the electronic tag circuit to send out the internal unique identification (UID) code or the stored data. At this point, the reader sequentially receives and interprets UID codes or data, which are sent to the computer application database for processing. The electronic tag itself does not need batteries and can work indefinitely. The plan is to apply the RFID system on a blind guide robot and to install an electronic tag on a guide brick, enabling the reader installed at the bottom of the blind guide robot to read the

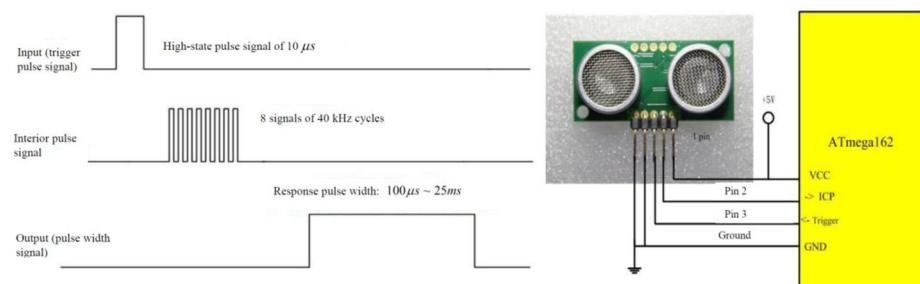


Figure 6.
Ultrasonic sensor
control timing and
pin signal diagram

UID code on the electronic tag when passing by the guide brick. Then the microcontroller ATmega162 current interprets the location or landmark and informs the blind person with a voice broadcast to decide the next direction for the wheeled mobile robot to go. [Figure 7](#) shows the system's reader and electronic tag, which are produced by Parallax Company.

The RFID reader used in this system works under 5 V of DC voltage. As long as the /ENABLE pin is given a low potential energy, the reader can read the UID code within the approaching electronic tag. When the reader senses the electronic tag, the electronic tag will send out 12 bytes of ASCII identification code at the transmission rate of 2,400 bps. The reader can read the UID identification code on the electronic tag. UID identification code is composed of ASCII code with 12 bytes. The starting bit is (0 × 0A), and the end bit is (0 × 0D), indicating that the string of data is valid. The intermediate 10 bytes are the UID identification code of the electronic tag, as shown in [Figure 8](#).

2.1.4 Voice broadcast system. The system uses the ISD4004 voice chip introduced by the USA ISD Company ([ISD, 2000](#)). The chip can repeatedly record, playback and store voice data longer and has a simple peripheral circuit, no additional memory, etc. We connected the voice chip to ATmega162 microcontroller to provide the blind person a voice broadcast system. [Figure 9](#) is the actual wiring diagram of a voice chip. The voice chip uses SPI synchronous serial communication interface as the transmission of data or instruction code.

2.2 The image capturing control unit

2.2.1 CMOS image sensor. This paper used the CMOS image sensor CMUCam3 ([Nourbakhsh, 2007](#)) produced by Acroname Company. The development core of CMUCam3 is based on ARM7TDMI, which is integrated with the embedded system and not only reduces the volume and improves the image sensing performance, but is also equipped with OV6620 and OV7630 full wide-angle CMOS cameras as well as expansion memory AL422B of 3M-bits stored pixel data, power supply and RS232 serial communication interface. As shown in [Figure 10](#), CMUCam3 not only has the ability to make the colors even and identify

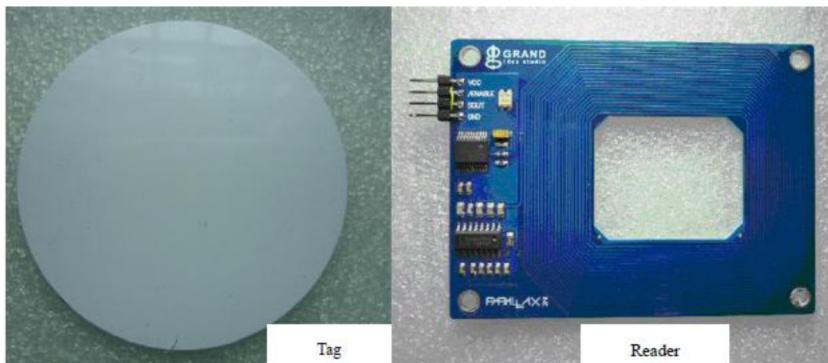


Figure 7.
Card reader and
electronic tag

MSB	Unique ID Digit 1	Unique ID Digit 2	Unique ID Digit 3	Unique ID Digit 4	Unique ID Digit 5	Unique ID Digit 6	Unique ID Digit 7	Unique ID Digit 8	Unique ID Digit 9	Unique ID Digit 10	LSB
Start Byte (0xA)											Stop Byte (0xD)

Figure 8.
UID code 12 bytes
data transmission
format

Figure 9.
Actual wiring
diagram of voice chip

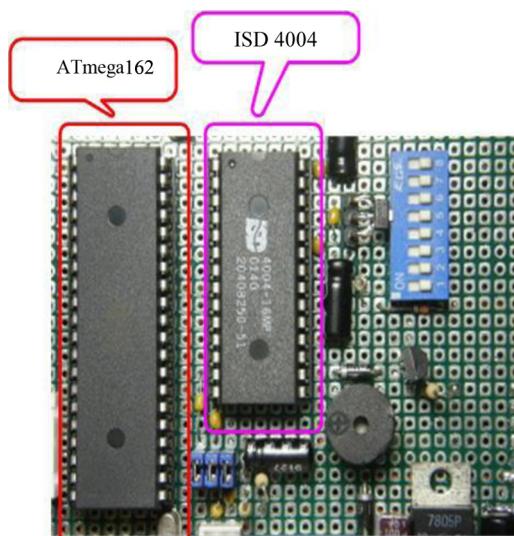
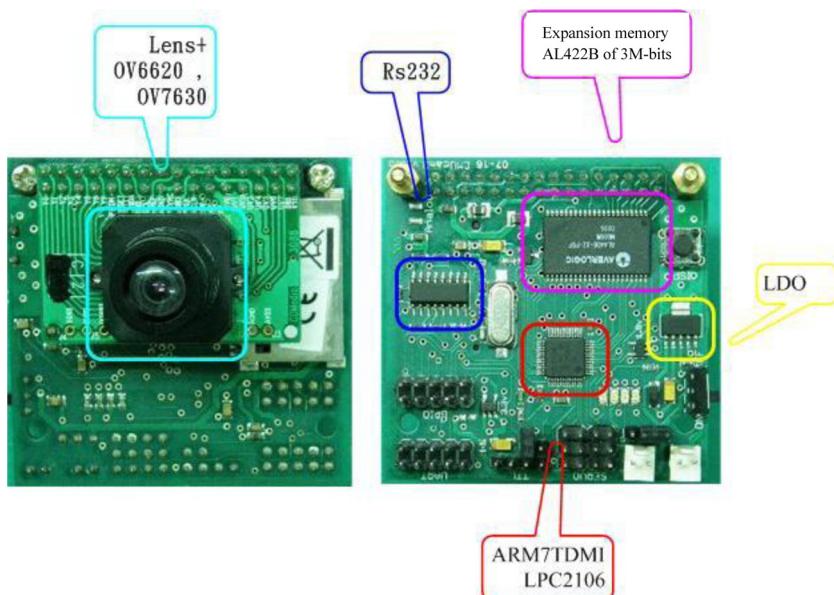


Figure 10.
CMUcam3 image
sensor entity
elements diagram



chromatic aberrations, but also has the functions of color movement and object movement tracking. Through an adjustable focal length of the lens, the quality of output image can be adjusted, and any size of the image window can be captured. In addition, CMUcam3 module I/O ports support the servo motor drive control to facilitate user tracking control and other

applications. The maximum image resolution is 352×288 pixels, and users can choose from three kinds of video output formats: RGB, CrYCb and HSV. The image resolution of this system uses 128×64 pixels. The video output format is RGB. In addition, the DC voltage of the module work has a very wide range; 9 ~ 12 VDC is acceptable.

Figure 11 is the 128×64 picture as video output. The packet format is sent out by CMUcam3. The start code is 01H. The two bytes appearing after that represent the size of the x-axis pixel and y-axis pixel of this image picture. After that, the y number of 02H will appear as R, G and B image data of the start code, with a total of 128 RGB bytes. R, G and B have three bytes, so after the start code 02H, 128×3 bytes will appear, with a total of 384 bytes. Above 384 + 1 bytes will repeatedly appear 64 times. That is $385 \text{ bytes} \times 64$, a total of 24,640 bytes. Finally, 03H, which is the end code, appears. Adding the whole start code and end code, the completely packed data will have a total of $1 + 2 + 64 \times (1 + (128 \times 3)) + 1 = 24,664$ bytes.

2.2.2 Image conversion and two-dimensional plane coordinates. The color image information captured by the CMOS image sensor goes through grayscale image and binarization processing. Each pixel is stored in the two-dimensional array data structure. Take for example the 128×64 pixel color film. If the start code and end code are deducted, the complete packet data total $64 \times (128 \times 3) = 24,576$ bytes. After grayscale conversion, the image data are a total of $64 \times 128 = 8,192$ bytes. Then after binarization processing, the image data had 1,024 bytes left. Finally, the binary image was converted and compressed into 16×8 plane coordinates to comply with the format of the drawing-type liquid crystal display. In other words, each coordinate point constitutes of 8×8 bits (8 bytes) in data format, as shown in **Figure 12**.

2.2.3 Data fitting algorithm. Considering there are N data points $(x_i, y_i), i = 1, 2, \dots, N$ on a two-dimensional coordinate plane, we used a straight line $y(x)$ for fitting data to approach the N data points (Eberly, 1998; Coope, 1993; Duda and Hart, 1973). This straight line can be expressed as:

$$y(x) = y(x : m, c) = mx + c \quad (3)$$

To make this straight line approach the said N data points as much as possible, we used the chi-square cost function as the measured message. The chi-square cost function can be written as:

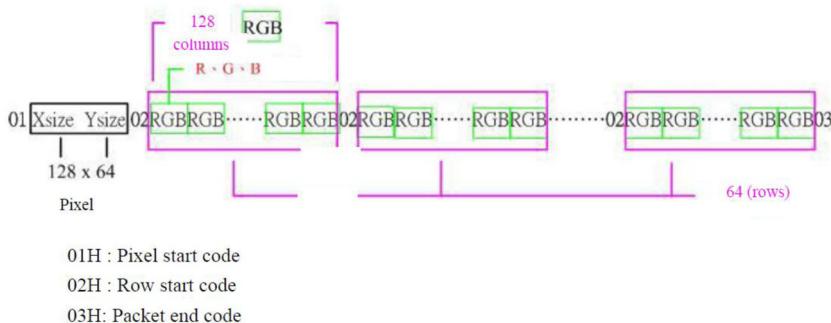


Figure 11.
CMOS image output
specifications

$$\chi^2(m, c) = \sum_{i=1}^N \left(\frac{y_i - c - mx_i}{\sigma_i} \right)^2 \quad (4)$$

We hoped to obtain a set of (m, c) , as the smaller the chi-square cost function, the better. To obtain the minimum value of [equation \(4\)](#) chi-square evaluation function, we conducted partial differentiation of [equation \(4\)](#), respectively, for the two variables m and c , and the result was zero. Then simultaneous equations were solved to obtain (m, c) . The results after partial differentiation are shown below:

$$0 = \frac{\partial \chi^2(m, c)}{\partial c} = -2 \sum_{i=1}^N \frac{y_i - c - mx_i}{\sigma_i^2} \quad (5)$$

$$0 = \frac{\partial \chi^2(m, c)}{\partial m} = -2 \sum_{i=1}^N \frac{x_i(y_i - c - mx_i)}{\sigma_i^2} \quad (6)$$

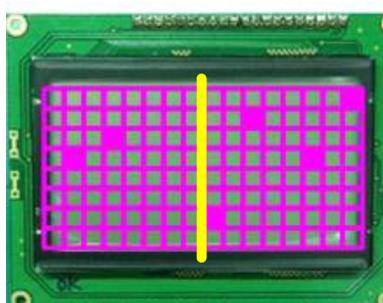
To make [equations \(5\)](#) and [\(6\)](#) look more concise and clear, we defined the following expressions:

$$S \cong \sum_{i=1}^N \frac{1}{\sigma_i^2}, S_x \cong \sum_{i=1}^N \frac{x_i}{\sigma_i^2}, S_y \cong \sum_{i=1}^N \frac{y_i}{\sigma_i^2}, S_{xx} \cong \sum_{i=1}^N \frac{x_i^2}{\sigma_i^2}, S_{xy} \cong \sum_{i=1}^N \frac{x_i y_i}{\sigma_i^2}, \quad (7)$$

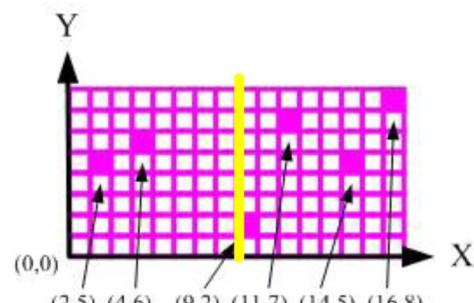
With the above expression (7), expressions (5 and 6) can be rewritten as:

$$\begin{cases} cS + mS_x = S_y \\ cS_x + mS_{xx} = S_{xy} \end{cases} \quad (8)$$

Simultaneous equations of solution (8) can obtain the following results:



(a)



(b)

Figure 12.
The image displays and plane coordinates conversion diagram

$$c = \frac{S_{xx}S_y - S_xS_{xy}}{\Delta}, m = \frac{SS_{xy} - S_xS_y}{\Delta} \quad (9)$$

Wherein $\Delta = SS_{xx} - (S_x)^2$, m is the slope of the linear equation, and c is the intercept of the linear equation.

2.2.4 CMOS image hunt strategy.

- The image picture of 128×64 pixels after grayscale processing and binarization was divided into left (1) and right (2) intervals of 64×64 pixels, as shown in [Figure 12](#).
- The coordinate position (x_i, y_i) and number of coordinate points c_1 and c_2 of left (1) and right (2) plane intervals were obtained.
- The data fitting algorithm was conducted to obtain left slope m_1 and right slope m_2 of the left (1) and right (2) plane intervals. The left slope m_1 and right slope m_2 can be used as the basis for comparison of moving the wheeled mobile robot in a straight line.
- After a data comparison, the right way forward and rotation angle θ is done. (The red arrow is the direction of original movement, and the blue dashed arrow is the correct direction of moving.)
- The left slope m_1 data are compared with the straight moving data to get angle α . This angle is also the angle θ that the wheeled mobile robot should amend (θ_m is the angle converted from the slope of marking when the robot is moving straight in the middle of the marking, and θ_n is the angle converted from the left slope m_1 to $\theta = \theta_m - \theta_n$).

2.2.4.1 Situation 1: the wheeled mobile robot is moving straight in the middle of the marking. The image picture captured by the COMS image sensor is as shown in [Figure 13](#). The wheeled mobile robot moves forward in the middle of the track. In the left (1) and right (2) blocks, the number of coordinate points searched is $c_1 = c_2 > 0$, and the pixel coordinates of the left (1) and right (2) blocks through the data fitting algorithm can obtain the left slope $m_1 > 0$, right slope $m_2 < 0$ and $|m_1| \cong |m_2|$. The data comparison results showed that the moving direction of the wheeled mobile robot is $\theta = \theta_1 = 0^\circ$.

2.2.4.2 Situation 2: the wheeled mobile robot leans extremely to the right. The image picture captured by the COMS image sensor is as shown in [Figure 14](#). As the wheeled mobile robot leans entirely to the right, the CMOS sensor lens can only shine on the right side of the track marking but cannot shine on the left side of the marking. After image pre-processing,

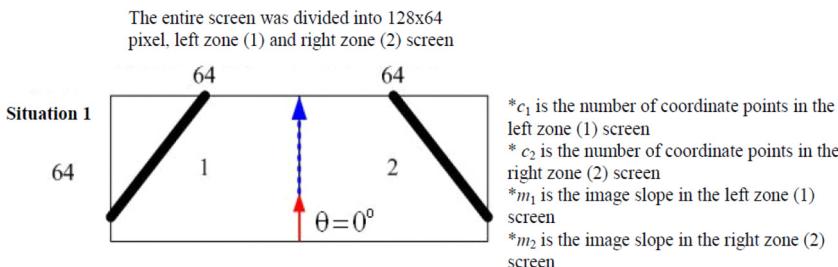


Figure 13.
Image picture of
Situation 1

the left (1) and right (2) blocks searched and found the number of coordinate points $c_1 = 0$ and $c_2 > 0$. Through the data fitting algorithm, the pixel coordinates that are fitted in a straight line can calculate that the left slope does not exist (no coordinate points), while the linear slope of the right interval is $m_2 < 0$. After the data comparison, the wheeled mobile robot got the correct direction of moving forward at $\theta = \theta_{2L}^o$.

2.2.4.3 Situation 3: the wheeled mobile robot leans extremely to the left. As the entire wheeled mobile robot leans to the left, the CMOS image sensor lens can capture the left track marking but not the right marking. Therefore, after the image pre-processing, the number of coordinate points $c_1 > 0$ and $c_2 = 0$ was searched and found in the left (1) and right (2) blocks. The data fitting algorithm calculated the left slope $m_1 > 0$, and the linear slope of the right interval was not present (no coordinate points). After data comparison, the wheeled mobile robot got the correct direction of moving at $\theta = \theta_{3R}^o$ (Figure 15).

2.2.4.4 Situation 4: the wheeled mobile robot leans slightly to the right. In the situation of the wheeled mobile robot leaning slightly to the right, the CMOS image sensor's camera angle not only covers the track marking on the right, it even captures a part of the track marking on the left. Figure 16 is an image picture captured by the CMOS sensor. At this time, both the left (1) and right (2) intervals have coordinate points, and the numbers, respectively, are $c_1 > 0$ and $c_2 > 0$; moreover, $c_1 < c_2$. The slopes of the track marking at the left (1) and right (2) intervals were, respectively, calculated, and $m_1 > 0$ and $m_2 < 0$ were obtained; moreover, $|m_1| < |m_2|$. After data comparison, the wheeled mobile robot got the proper angle of moving at $\theta = \theta_{4L}^o$.

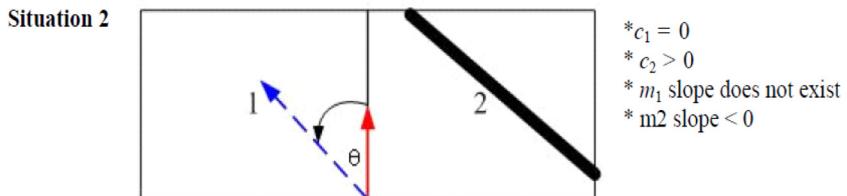


Figure 14.
Image picture of
Situation 2

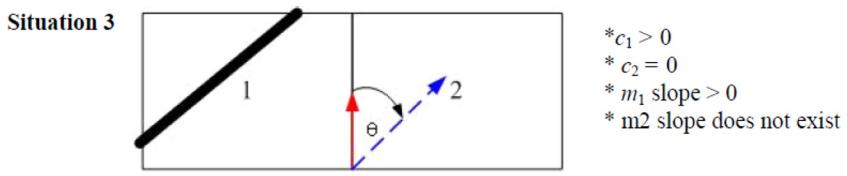


Figure 15.
Image picture of
Situation 3

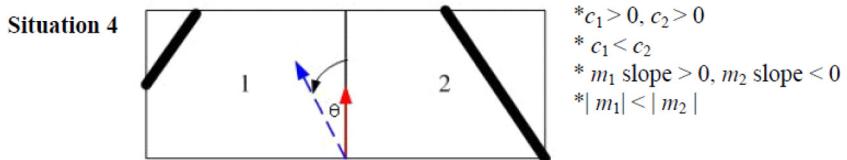


Figure 16.
Image picture of
Situation 4

2.2.4.5 Situation 5: the wheeled mobile robot slightly leans to the left. Figure 17 shows the image picture captured by the CMOS image sensor and shows the situation of the wheeled mobile robot slightly leaning to the left. The image captured covers track marking on the left and a small part of the track marking on the right. The numbers of coordinate points in both left (1) and right (2) intervals, respectively, are $c_1 > 0$ and $c_2 > 0$; moreover, $c_1 > c_2$. The slopes of marking at left (1) and right (2) intervals are $m_1 > 0$ and $m_2 < 0$, respectively; moreover, $|m_1| > |m_2|$. After data comparison, the wheeled mobile robot got the proper angle of moving at $\theta = \theta_{5R}^o$.

2.2.4.6 Situation 6: the wheeled mobile robot leans a lot to the right. Finally, we discussed two special conditions. The wheeled mobile robot leans extremely to the right and leans extremely to the left (Figure 18). The angle of the robot leaning to the right is too large. The CMOS image sensor lens covers the range to the right of the track marking too long, extending to the block of left (1) interval. At this time, both left (1) and right (2) intervals have coordinate points, wherein $c_1 > 0$ and $c_2 > 0$; moreover, $c_1 < c_2$. The slopes of marking appearing in the left (1) and right (2) intervals were used to calculate and obtain $m_1 < 0$ and $m_2 < 0$. After data comparison, the wheeled mobile robot got the proper angle of moving at $\theta = \theta_{6L}^o$.

2.2.4.7 Situation 7: the wheeled mobile robot leans a lot to the left. In Situation 7, the angle of the wheeled mobile robot leaning to the left is too large, resulting in the CMOS image sensor lens covering more than the left track marking range and extending to the right block (2) (Figure 19). At this time, both left (1) and right (2) intervals have coordinate points, wherein $c_1 > 0$ and $c_2 > 0$; moreover, $c_1 > c_2$. The marking slopes appearing in the left (1) and right (2) intervals can be calculated as $m_1 > 0$ and $m_2 > 0$, respectively. After comparison, the wheeled mobile robot got the correct angle of moving at $\theta = \theta_{7R}^o$ (Table I).

3. The charging control unit

The battery charge system designed in this system uses a low-cost AVR 8-bit microcontroller ATtiny25 (Atmel, 2013a, 2013b) as the core unit of the system. The small size has only eight pins. ATtiny25 contains 2 K-byte Flash ROM, 128-byte SRAM, 128-byte EEPROM memory, 10-bit A/D converter, PWM output function and Buck conversion switch

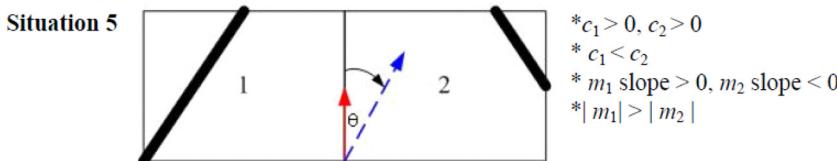


Figure 17.
Image picture of
Situation 5

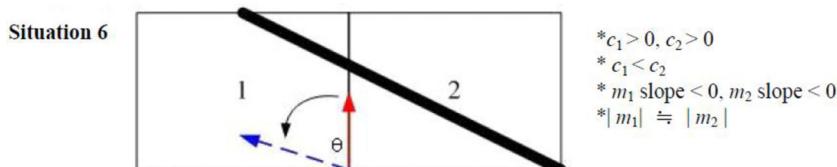


Figure 18.
Image picture of
Situation 6

circuit to achieve the purpose of the battery charger. Furthermore, ATtiny25 internal EEPROM memory cell can be used to save the parameters of the battery characteristics, such as charging records, to control the battery capacity actually used. The 10-bit A/D converter can provide sufficient accuracy, making the fully charged battery closer to its maximum capacity. The battery charger circuit uses the PWM signal to control the Buck conversion switch circuit for charging and discharging action. The internal 10-bit A/D converter is used to monitor the battery voltage, charging voltage and charging/discharging current value. Figure 20 shows the ATtiny25 battery charger block diagram. In the program, we first set, respectively, the input supply voltage maximum value as 9 V, the maximum charging current as 1 A and the battery charging voltage as 8.4 V ($4.2\text{ V} \times 2$). The battery in series is 1S2P.

The control core of the ATtiny25 battery charger uses the communications protocol of 9,600 bps, 8 data bits, no parity and 1 stop bit to transmit the characteristic parameters and situation of the battery at any time to the main system's microcontroller ATmega162. The total number of transmitted data of battery situations is 24 bytes. We have custom-made complete data formats. In every piece of data, a fixed header and ending are added to facilitate identification of the main system microcontroller. The formats of the header and ending are shown in Table II.

3.1 ADC battery charger measure circuit

The single cell voltage is V_1 , the battery voltage is V_2 and the charging voltage is V_3 , which can be measured directly from the battery (Figure 20). The ATtiny internal 10-bit ADC was used to measure the charging voltage. If it is higher than the reference voltage (V_{cc}), the two resistors need to be increased to make the voltage division reach the $0 \sim V_{cc}$ range. As for the charging current, it was obtained by subtracting the battery voltage V_2 from the charging voltage V_3 and then dividing the result by a very small precision resistor (R_{sense}).

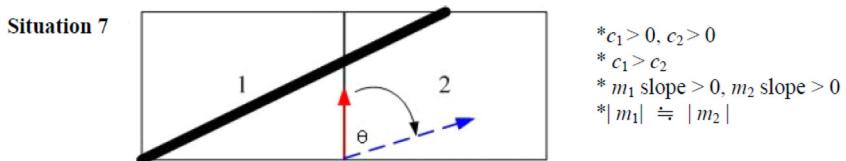
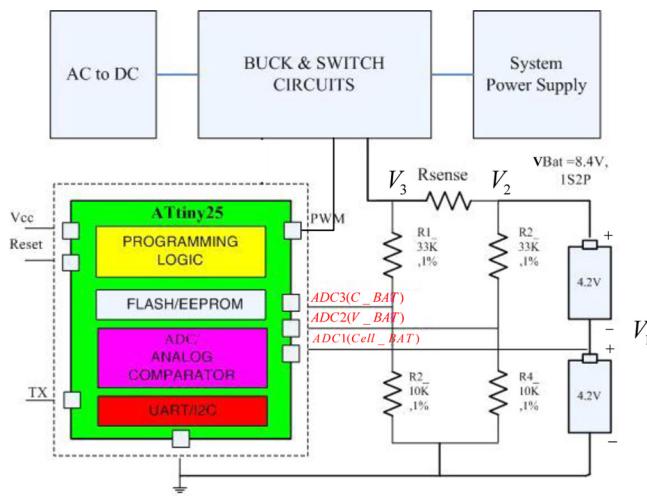


Figure 19.
Image picture of
Situation 7

Table I.
The wheeled mobile
robot's hunt strategy
angle correction table

Situation	No. of coordinate points in the left interval (C_1)	No. of coordinate points in the right interval (C_2)	The left-interval pixel slope (m_1)	The right-interval pixel slope (m_2)	Leaning angle ($\theta = \theta_m - \theta_n $)
1	$C_1 > 0$	$C_2 > 0$	$m_1 > 0$	$m_2 < 0$	$\theta = 0^\circ$
2	$C_1 = 0$	$C_2 > 0$	None	$m_2 < 0$	$\theta = \theta_{2L}$
3	$C_1 > 0$	$C_2 = 0$	$m_1 > 0$	none	$\theta = \theta_{3R}$
4	$C_1 > 0$	$C_2 > C_1$	$m_1 > 0$	$m_2 < 0$	$\theta = \theta_{4L}$
5	$C_1 > C_2$	$C_2 > 0$	$m_1 > 0$	$m_2 < 0$	$\theta = \theta_{5R}$
6	$C_1 > 0$	$C_2 > C_1$	$m_1 < 0$	$m_2 < 0$	$\theta = \theta_{6L}$
7	$C_1 > C_2$	$C_2 > 0$	$m_1 > 0$	$m_2 > 0$	$\theta = \theta_{7R}$


Figure 20.
ATtiny25 battery
charger block
diagram

Data name	Data patterns	Values	No.	Explanation
HEAD	Unsigned char	$0 \times 5B$	1	Header
Message situation	Unsigned char		16	Battery situation information
VBAT, M_BAT, IBAT	Unsigned char		6	Parameter information
END	Unsigned char	$0 \times 0A, 0 \times 0D$	2	Ending

Table II.
Battery situation
data transfer format
specifications

In other words, in addition to generating PWM signals, the ATtiny25 microcontroller in **Figure 20** is also responsible for the conversion of the three groups of ADC. ADC3 is the analog voltage value of voltage division conducted on the charging voltage V_3 by capture of R_1 and R_2 and is also called C_BAT. ADC2 is the analog voltage value of voltage division conducted on battery voltage V_2 by capture of R_3 and R_4 and is also called V_BAT. ADC1 is the direct capture of single cell voltage V_1 value and is also known as Cell_BAT. As the single cell voltage V_1 value is less than the reference voltage of the system, it does not require an additional dividing circuit. R sense resistance value is usually a very small precision resistance. If the charging voltage V_3 minus the battery voltage value V_2 and then divided by R sense resistance is equal to the positive current value, it is known as the battery charging current. Conversely, if it is equal to the negative current value, it is known as the battery discharge current.

3.2 Battery voltage and charging voltage

In design, if $AREF = Vcc = 5V$ is the reference voltage, the ADC measurement range is $AGND \sim AREF (5V)$. V_3 is the output voltage of the Buck conversion switch circuit. The dividing circuit design adopts $R_1 = 33K\Omega$ and $R_2 = 10K\Omega$. The maximum input voltage for input into the ADC3 terminal is the reference voltage. The group of voltage dividers on the left of **Figure 20** can reach the maximum charging voltage:

$$V_3 = \left(\frac{R_1 + R_2}{R_2} \right) \times ADC3 = \left(1 + \frac{33K}{10K} \right) \times 5V = 21.5V \quad (10)$$

In the design of the system, the battery charging voltage value must be fixed to be equal to the maximum value of the battery voltage, i.e. the Buck conversion switch circuit output voltage $V_3 \leq 8.4V$ meets the demand of the maximum charging voltage of 21.5 V. Similarly, the output voltage of the group of voltage dividers on the right side of [Figure 20](#) was obtained by using the battery voltage V_{BAT} to conduct voltage division on R_3 and R_4 and to directly connect to input pin ADC2 of the ATtiny25 internal ADC converter. The maximum battery voltage V_{BAT} can also reach up to 21.5 V. However, we only concatenated two 4.2-V single cells to combine them to make 8.4 V and to fully meet the needs of the design.

3.3 Charging current value

The measurement of the charging current I_{BAT} was achieved by passing through the precision minimal resistance value of 0.25 ohm. It was obtained by the charging voltage V_3 minus the battery voltage value V_2 and divided by Rsense resistance. If the result is equal to a positive current value, it is referred to as the battery charging current. Conversely, if the result is equal to a negative current value, it is referred to as battery discharge current. As for the battery charge/discharge current, it can be expressed as:

$$\begin{aligned} I_{BAT} &= \frac{V_3 - V_2}{R_{sense}} = \frac{1}{R_{sense}} \left[\left(\frac{R_1 + R_2}{R_2} \right) \times ADC3 - \left(\frac{R_3 + R_4}{R_4} \right) \times ADC2 \right] \\ &= \frac{1}{R_{sense}} \left[\left(\frac{R_1 + R_2}{R_2} \right) \times (ADC3 - ADC2) \right] \end{aligned} \quad (11)$$

The relationship among charging current I_{BAT} , charging voltage ADC3 and battery voltage ADC2 can be re-written as:

$$\begin{aligned} ADC3 - ADC2 &= \left(\frac{R_2}{R_1 + R_2} \right) \times I_{BAT} \times R_{sense} = \left(\frac{10K}{33K + 10K} \right) \times I_{BAT} \times 0.25 \\ &= 0.058 \times I_{BAT} \end{aligned} \quad (12)$$

Both the ADC2 and ADC3 values are similar. In other words, the difference between the two is close to zero. To avoid insufficient accuracy of ATtiny25 internal A/D converter, we added an external amplifier gain of 20 folds, so that [equation \(12\)](#) becomes:

$$ADC4 = 20 \times 0.058 \times I_{BAT} = 1.16I_{BAT} \quad (13)$$

ADC4 is the difference between ADC2 and ADC3 multiplied by a gain value of 20 times. Therefore, according to the design of the charging battery circuit in [Figure 20](#), the maximum charging current that the system can measure is:

$$I_{BAT(max)} = \frac{ADC4}{1.16} = \frac{5V}{1.16} = 4.31A \quad (14)$$

This meets the specification requirements of the maximum charging current of 1A as designed for this circuit.

4. Experimental results

4.1 CMOS image sensor verification test

The total size of the image buffer zone is $32K + 32K$ bytes. It stores 24,664 bytes of original images data, 7,936 bytes of grayscale data, 1,024 bytes of binary data and data after compression quantization. [Figure 21](#) is the schematic diagram that shows the use of two UART sets inside the ATmega162 microcontroller that are, respectively, connected by the RS232 level conversion chip to the PC and CMUcam3 image sensor modules. The ATmega162 microcontroller is used as the interface to receive/send and conduct verification and testing of CMUcam3 instructions.

The purpose of this experiment was to validate the results of the image pre-processing and plane coordinates after conversion. The 128×64 drawing-mode LCD was used to display the image after completion of processing on the screen. The CMOS image sensor was used to capture an image picture with two markings. The size was 128×64 pixels, as shown in [Figure 22\(a\)](#). After the grayscale processing, the RGB color picture was converted into a $0 \sim 255$ grayscale picture, as shown in [Figure 22\(b\)](#). Then through binarization processing, the $0 \sim 255$ grayscale picture was converted into a black-and-white picture with 0s and 1s only, as shown in [Figure 22\(c\)](#). As the binarized image before filtering was influenced by a number of noises in the determination of two marking locations, the binarization filter (adjust threshold) can remove most of the noises. The results of the filter are shown in [Figure 22\(d\)](#). Of course, the image after filtration will produce differences from the original. Some images will produce distortion and deformation. From the results it can be observed that the degree of distortion of the image compared to the original one is insignificant and within an acceptable range that does not affect judgment of markings on the guide brick.

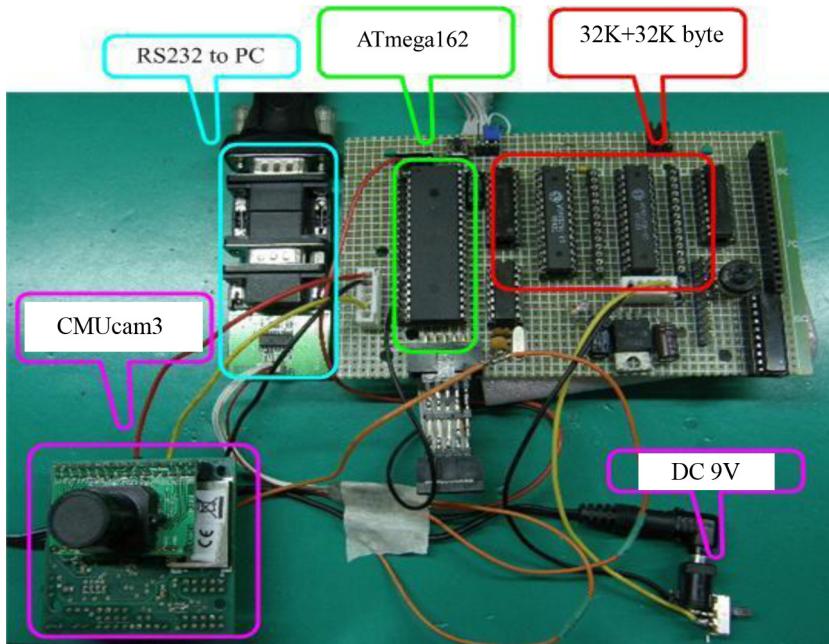


Figure 21.
CMOS image sensor
verification wiring
diagram

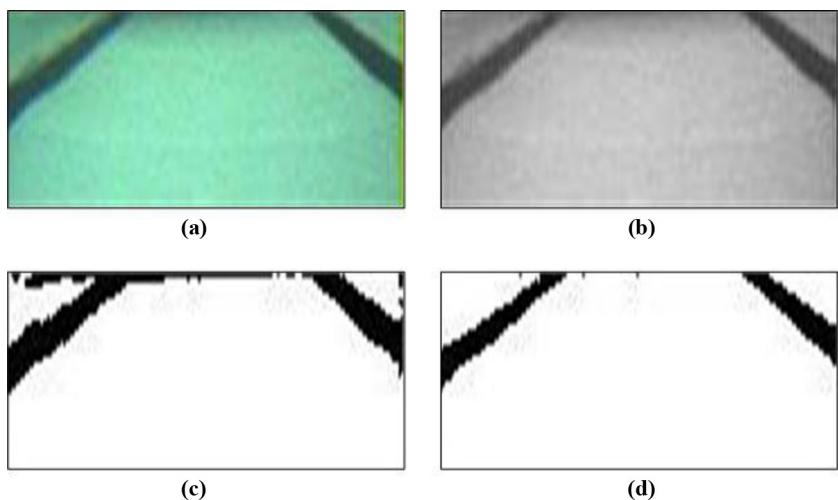


Figure 22.
Results of
binarization filter in
image processing

Finally, we wanted to convert the picture of binarization image of 128×64 pixels into 16×8 plane coordinates, as shown in Figure 23(a). The coordinate point of each unit consisted of 8×8 pixels. For the 64 pixels, if it is determined that more than half of the pixels (32) have “1” as the foreground. If, the unit coordinates are identified as “1”, namely, the image of the marking.

We then separated the 16×8 plane coordinates into two 8×8 plane coordinates. The coordinate values of each unit on the left and right markings are detected by the CMOS image sensor, as shown in Figure 24. Second, the data fitting algorithm was used to calculate, respectively, the slope of the markings of the left and right plane intervals, as shown in Figure 25. The CMOS image sensor captured a 128×64 -pixel color picture. After going through grayscale conversion, binarization processing, image compression, segmentation and the data fitting algorithm, the whole picture will be split, respectively, into two plane intervals and divided into coordinates of an 8×8 unit. The slopes m_1 and m_2 of the left and right markings and other important parameter values were also obtained. Then through the comparison operation, the correct angle θ was obtained, and correction of the advance angle of the wheeled mobile robot was conducted.

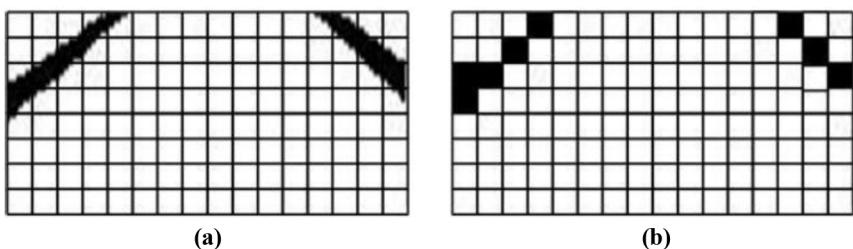


Figure 23.
Plane coordinates
conversion

4.2 The wheeled mobile robot moving straight and stopping

The wheeled robot was set in the center of the marking, as shown in [Figure 26\(a\)](#) below. At the same time, the RFID electronic tag was placed in the center of the marking, as shown in [Figure 26\(b\)](#). The image picture captured by the CMOS image sensor went through the image processing algorithms presented in this paper. The whole picture was divided into two 8×8 plane coordinates. The left-side plane marking coordinates obtained were $(1, 5), (2, 6), (3, 7)$ and $(4, 8)$. The number of coordinate points was $c_1 = 4$. Then the data fitting algorithm was used to obtain the left slope $m_1 = 1$ and $\theta_1 = 45^\circ$. Similarly, the right-side plane marking coordinates obtained were $(5, 8), (6, 7), (7, 6)$ and $(8, 5)$. The number of coordinate points is $c_2 = 4$, the right slope is $m_2 = -1$ and $\theta_2 = -45^\circ$. By calculation from the left interval slope $m_1 = 1$ and $\theta_1 = 45^\circ$ ($\theta_n = \theta_1$), we obtain the advance angle correction as $\theta = |\theta_m| - |\theta_n| = 0^\circ$. From [Table I](#), it can be learned that the wheeled mobile robot moved forward straight. When the electronic tag was placed in the center of the marking, the tag information was read by the RFID reader device installed under the wheeled mobile robot. The wheeled mobile robot immediately halted and started the voice broadcast.

4.3 The wheeled mobile robot leaning to the left

The wheeled mobile robot leans to the left of the marking, as shown in [Figure 27\(a\)](#) below. [Figure 27\(b\)](#) shows the picture of the robot continuing to advance after amendment of the angle. As the wheeled mobile robot is leaning entirely to the left, the CMOS sensor lens that captured

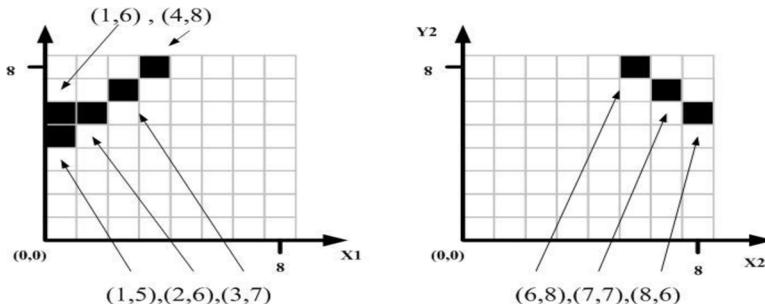


Figure 24.
Two 8×8 plane
coordinates and
marking image
coordinates

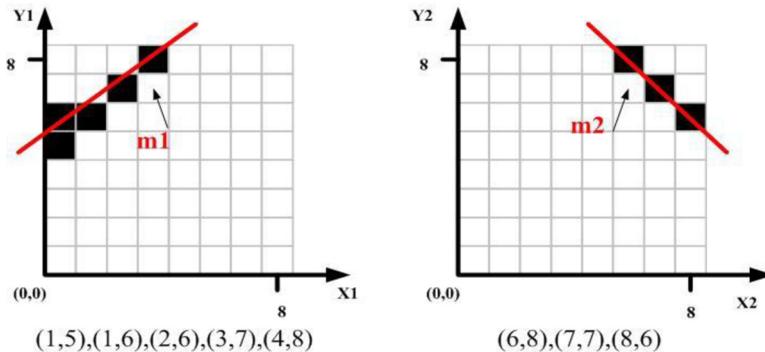
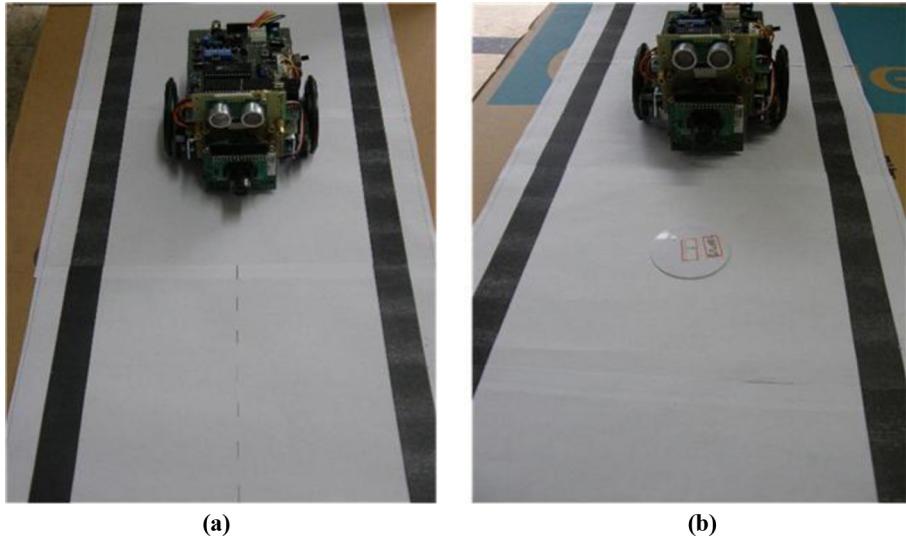


Figure 25.
The slope of the two
markings in the left
and right plane
intervals

**Figure 26.**

Wheeled mobile robot located in the center of the marking

track markings on the left could not shoot the marking on the right side. After image pre-processing and binarization plane coordinate conversion, the numbers of coordinate points searched and found by the left (1) and right (2) blocks are $c_1 = 7 > 0$ and $c_2 = 0$, respectively. The left-side plane marking coordinates were obtained as (1, 4), (2, 5), (3, 5), (4, 6), (5, 6), (5, 7) and (6, 7). Through the data fitting algorithm, pixel point coordinates were fit into a straight line. It can be learned that because the right-side plane has no coordinate points, the slope does not exist. The slope of the left plane marking is $m_1 = 0.5882$ and $\theta_1 = \theta_n = 30.4655^\circ$. After data comparison, it was known $\theta = |\theta_m| - |\theta_n| = 45^\circ - 30^\circ = 15^\circ$, and the wheeled mobile robot has obtained amendments to the direction of moving at $\theta = \theta_{3R} = 15^\circ$.

4.4 The wheeled mobile robot leaning to the right

The wheeled mobile robot's position was leaning to the right of the marking. Figure 28(a) and (b) are pictures showing the robot moving on after correction of the angle. As the wheeled mobile robot is leaning entirely to the right, the lens of the CMOS sensor can only shoot the track marking on the right, but cannot shoot the marking on the left side. After image pre-processing and binarization plane coordinate conversion, the left (1) and right (2) blocks searched and found the number of coordinate points $c_2 = 7 > 0$ and $c_1 = 0$, respectively. The left plane marking's point coordinates obtained were (2, 8), (3, 8), (4, 7), (5, 7), (6, 6), (7, 6) and (8, 5). Through the data fitting algorithm, pixel point coordinates were fitted into a straight line. It can be learned that because the left-side plane has no coordinate points, the slope does not exist. The slope of the right plane marking is $m_2 = -0.4999$ and $\theta_2 = 27^\circ$. After data comparison, it was known that $\theta = |\theta_m| - |\theta_n| = 45^\circ - 27^\circ = 18^\circ$. The wheeled mobile robot has obtained amendments to the direction of moving at $\theta = \theta_{2L} = 18^\circ$.

4.5 The wheeled mobile robot turning left

Figure 29(a) below shows the situation the wheeled mobile robot encountered in the process of making a left turn. When the RFID electronic tag was placed at the center of the left turn marking, the RFID reader installed underneath the wheeled mobile robot

read the RFID tag information. The wheeled mobile robot immediately halted and started a voice broadcast, as shown in [Figure 29\(b\)](#). Subsequently, the wheeled mobile robot conducted a left turn action. When the action was completed, it then continued to move forward, as shown in [Figure 29\(c\) and \(d\)](#) below. Notably, if the wheeled mobile robot encountered a left turn situation in the process of moving, the CMOS image sensor in front would be in the non-marking state for a period. At this time, the wheeled mobile robot moved forward 18 cm, and the RFID reader started detection. If it read information to make a left turn from the RFID tag connotation, the wheeled mobile robot immediately halted and started a voice broadcast to notify the blind person to turn left. After the robot completed the left turn, it then continued forward. In case any RFID tag message could not be detected, the wheeled mobile robot stopped advancing.

4.6 The wheeled mobile robot turning right

The situation for the wheeled mobile robot to turn right was encountered in the process of moving, as shown in [Figure 30\(a\)](#) below. The RFID electronic tag was placed at the center of the right turn marking. The RFID reader installed underneath the wheeled mobile robot read the RFID tag information. The wheeled robot immediately halted and started a voice broadcast, as shown in [Figure 30\(b\)](#). Subsequently the wheeled mobile robot made a right

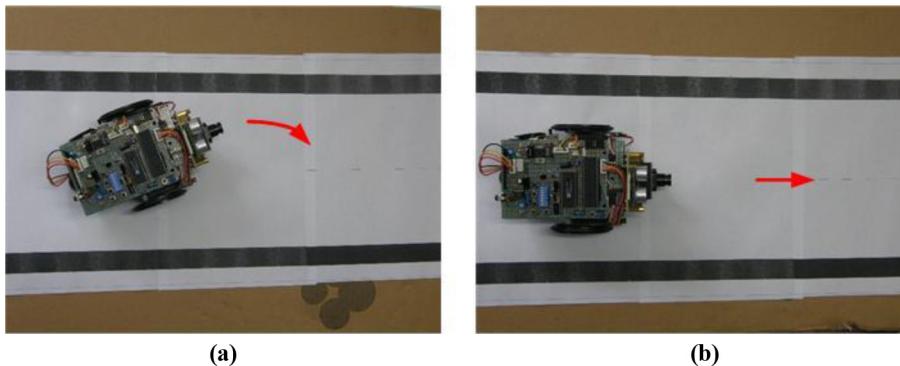


Figure 27.
Wheeled mobile
robot leaning to the
left (a) and moving
straight (b) after
correction

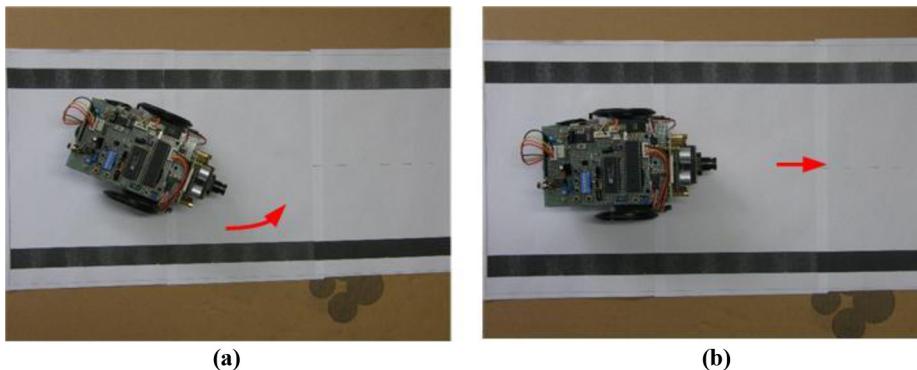


Figure 28.
Wheeled mobile
robot leaning to the
left (a) and moving
straight (b) after
amendment

turn. When it completed the action, it continued to move forward, as shown in [Figure 30\(c\) and \(d\)](#) below. Notably, if the wheeled mobile robot encountered during travel the situation for a right turn, the CMOS image sensor in front will be in the non-marking state for a period. Notably, if the wheeled mobile robot encountered a situation for a right turn in the process of moving, the CMOS image sensor in front will be in the non-marking state for a period. At this time, the wheeled mobile robot moved forward 18 cm and started the RFID reader for detection. If it read information to turn right from the RFID tag information, the wheeled mobile robot immediately halted and started the voice broadcast to notify the blind person to turn right. It then turned right and continued moving forward. If RFID tag messages could not be detected, the wheeled mobile robot stopped advancing.

5. Conclusions and future prospects

The main purpose of this paper was to establish a wheeled mobile robot on a working platform of the blind guide system and to integrate the increasingly sophisticated RFID system and voice broadcast system to read the hidden messages in the guide bricks or the Braille in public places. A CMOS image sensor detects two markings on the guide brick to achieve the goal design of blind guide robots. In the design, the image processing algorithms entirely achieved within the embedded microcontrollers unit, and the critical threshold value selection can only rely on the rough trial-and-error

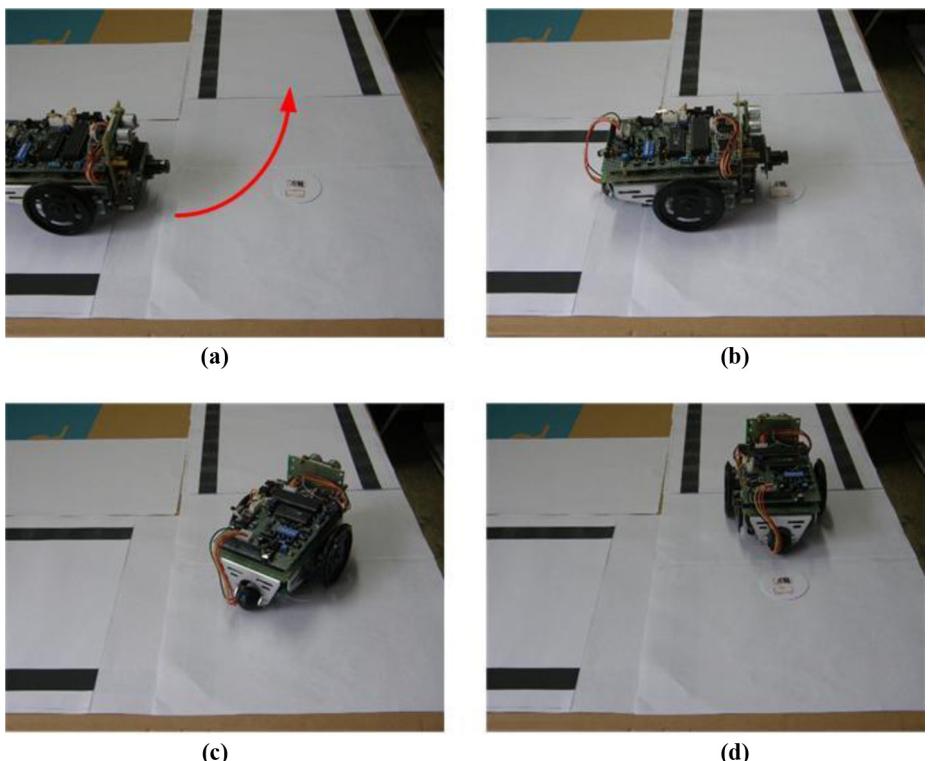


Figure 29.
The blind guide
robot turning left

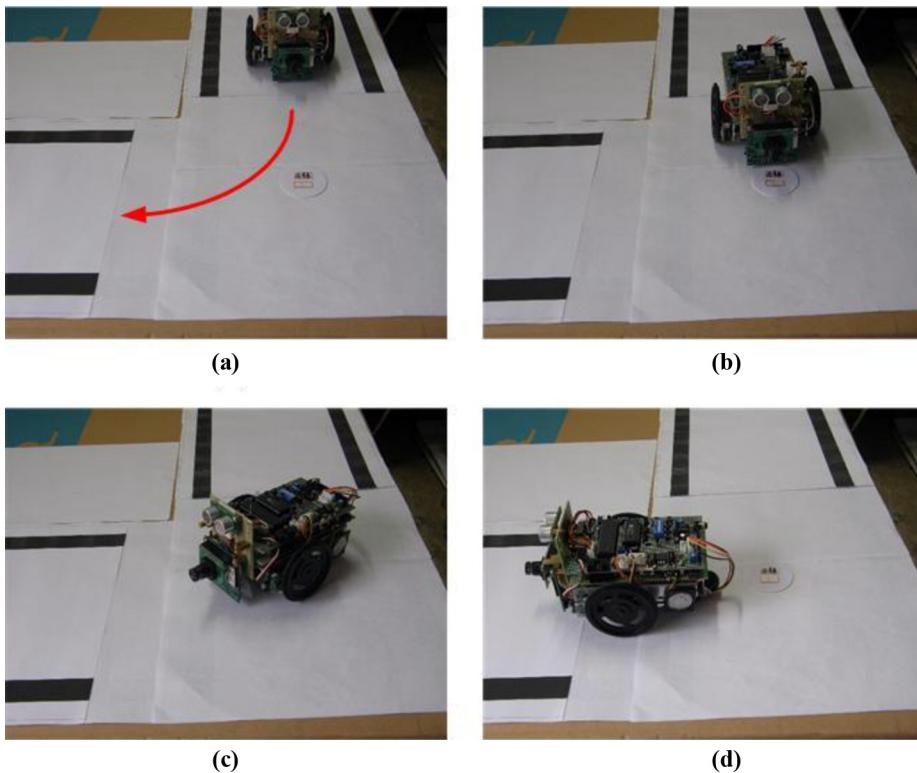


Figure 30.
Blind guide robot to
turn right

method to make decisions. When the images of the CMOS image sensor go through binarization processing, if the appropriate or optimal threshold value cannot be obtained, it easily leads to image distortion and algorithm errors. Second, in the RFID system, the electronic tag is pasted on the back of the guide brick. When the robot drives on the guide bricks, it can sense the UID code of the electronic tag and capture the message implied in the electronic tag. The current RFID system can only read the UID code of the electronic tag. The message contained is limited. In addition to UID code, in the future we also hope to read the memory contents of the electronic tag. By then the electronic tag will be able to load a larger amount of information, such as bus information, latitude and longitude coordinates of landmarks and so on. The system uses two 8-bit ATmega162 sets as the wheeled mobile robot's microcontroller:

- (1) one is used to control 2 RC SERVO elements on the body and the voice chip and RFID device; and
- (2) the other is used to capture image taken by the CMOS image sensor.

In future studies, the 32-bit embedded microprocessor ARM can be used as the controller of the main system, or an embedded microprocessor ARM 7 on CMOS image sensor can be used to first process the captured images through the algorithms and then transmit the useful information back to the ATmega162 microcontroller, which can reduce the heavy

computational load of the microcontroller. In addition, we hope to integrate the blind guide robot into the white cane to guide the blind, and to enhance the performance of the CMOS image sensors, which can be combined with wireless internet and satellite global positioning system to accurately return the current location and coordinates of the visually impaired and re-transmit the trajectory of the blind to a remote master computer for monitoring.

238

References

- Atmel Corporation (2013a), *Atmega 162/v Datasheet*, Atmel Corporation, pp. 1-324.
- Atmel Corporation (2013b), *A Ttiny 25/45/85 Datasheet*, Atmel Corporation, pp. 1-173.
- Borenstein, J. (2001), "The guidecane-applying mobile robot technologies to assist the visually impaired", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 31 No. 2, pp. 131-136.
- Coope, I.D. (1993), "Circle fitting by linear and nonlinear least squares", *Journal of Optimization Theory and Applications*, Vol. 76 No. 2, pp. 381-388.
- Devantech (2008), *SRF05 Datasheet*, Devantech, pp. 1-9.
- Duda, R.O., and Hart, P.E. (1973), *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, NY.
- Eberly, D. (1998), *Least Squares Fitting of Data*, Geometric Tools, pp. 1-9.
- Gupta, S., Sharma, I., Tiwari, A., and Chitranshi, G. (2015), "Advanced guide cane for the visually impaired people", 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, pp. 452-455.
- ISD (2000), *ISD4004 Series-Datasheet Catalog*, ISD pp. 1-29.
- Levine, S.P., Bell, D.A., Jaros, L.A., Simpson, R.C., Koren, Y. and Borenstein, J. (1999), "The NavChair assistive wheelchair navigation system", *IEEE Transactions on Rehabilitation Engineering*, Vol. 7 No. 4, pp. 443-451.
- Noetic Design (2005), *WheelWatcher WW-02 Product Manual: Incremental Quadrature Encoder System for Gearhead Motors*, Noetic Design, pp. 1-10.
- Nourbakhsh, I. (2007), *CMUcam3 Datasheet v1.02 for the CMUcam3 Embedded Vision Sensor*, Carnegie Mellon University, pp. 1-31.
- Polycarpou, A.C., Dimitriou, A., Bletsas, A., Polycarpou, P.C., Papaloizou, L., Gregoriou, G. and Sahalos, J.N. (2012), "On the design, installation, and evaluation of a radio-frequency identification system for healthcare applications (wireless corner)", *IEEE Antennas and Propagation Magazine*, Vol. 54 No. 4, pp. 255-271.
- Shoval, S., Borenstein, J. and Koren, Y. (1998), "The NavBelt-a computerized travel aid for the blind based on mobile robotics technology", *IEEE Transactions on Biomedical Engineering*, Vol. 45 No. 11, pp. 1376-1386.
- Shoval, S., Ulrich, I. and Borenstein, J. (2003), "NavBelt and the Guide-Cane (obstacle-avoidance systems for the blind and visually impaired)", *IEEE Robotics and Automation Magazine*, Vol. 10 No. 1, pp. 9-20.
- Sarma, E.S., Stephen, A.W., and Daniel, W.E. (2003), "RFID systems and security and privacy implications", *Cryptographic Hardware and Embedded Systems - CHES 2002*, pp. 454-469.

Corresponding author

Pu-Sheng Tsai can be contacted at: tps@ee2.cust.edu.tw

For instructions on how to order reprints of this article, please visit our website:

www.emeraldgroupublishing.com/licensing/reprints.htm

Or contact us for further details: permissions@emeraldinsight.com