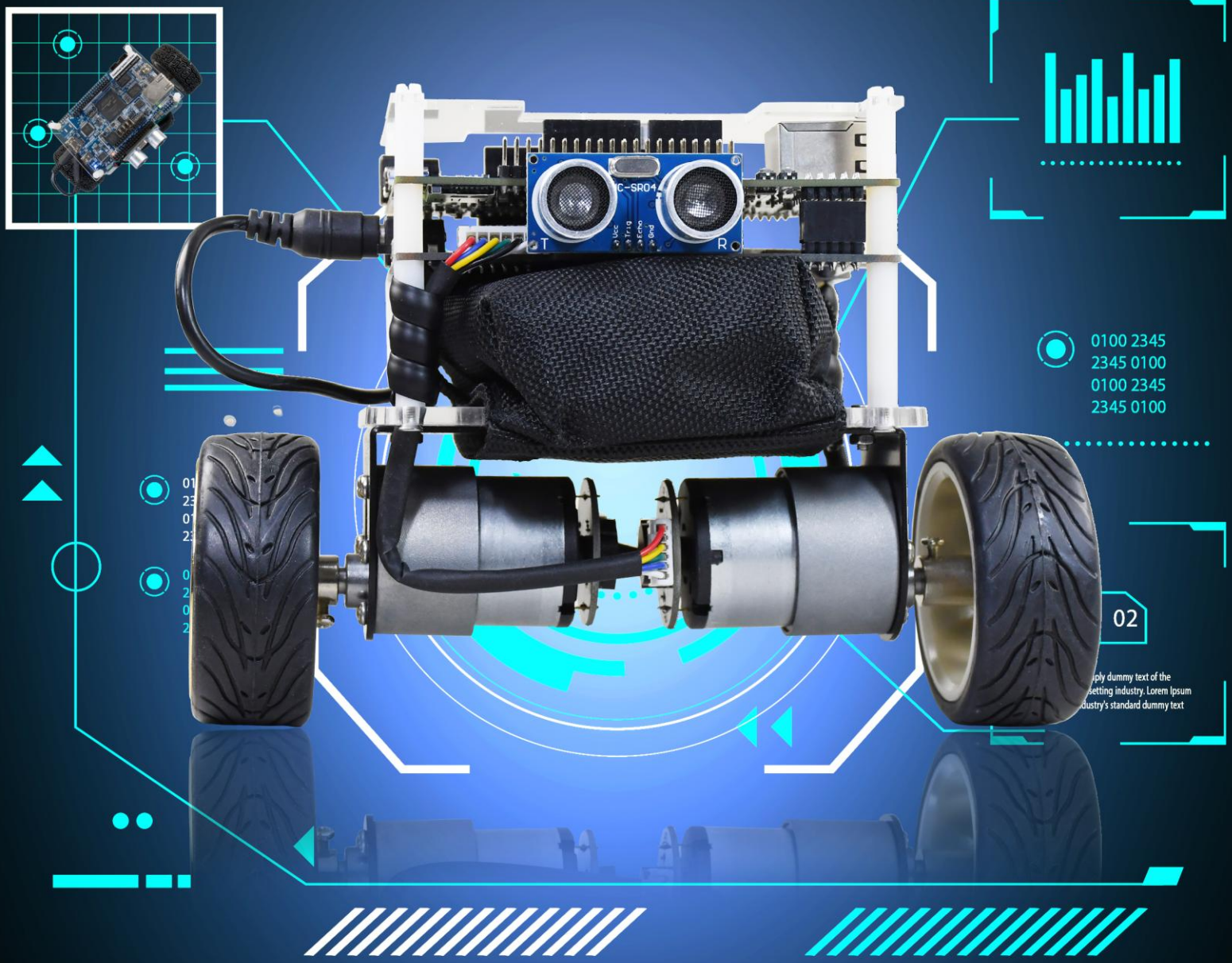
Self-Balancing Robot

Getting Started Guide



terasIC
www.terasic.com

Copyright © 2003-2018 Terasic Inc. All Rights Reserved.

CONTENTS

CHAPTER 1 *INTRODUCTION* 1

1.1 PACKAGE CONTENTS 1

CHAPTER 2 *COMPONENTS AND FUNCTIONS* 3

2.1 PARTS AND FUNCTIONS 3

2.2 DE10-NANO KIT AND MOTOR DRIVER BOARD 6

CHAPTER 3 *SETUP ELEMENTS* 1

3.1 CONFIGURATION MODE SWITCHES 1

3.2 OPERATION MODE SWITCHES 2

3.3 LEDS ON THE MOTOR DRIVER BOARD 3

3.4 LEDS ON DE10-NANO BOARD 4

CHAPTER 4 *BASIC OPERATIONS* 6

4.1 CONNECT POWER PORT 6

4.2 POWER ON THE ROBOT 7

4.3 KEEP A BALANCED STATE 8

4.4 ATTITUDE CONTROL 8

CHAPTER 5 *ADVANCED FEATURES DEMONSTRATION* 9

5.1 OBSTACLE AVOIDANCE DEMONSTRATE 9

5.2 SMARTPHONE APP CONTROL 11

5.3 IR REMOTE CONTROL 17

CHAPTER 6 *CHARGING THE BATTERY* 18

CHAPTER 7 *RESTORE FACTORY SETTING* 20

7.1 ARM VERSION RESTORING 20

7.2 NIOS VERSION RESTORING 22

ADDITIONAL INFORMATION 25

Chapter 1

Introduction

Based on the Terasic DE10-Nano SoC board platform, designed and manufactured independently by Terasic, the Self-Balancing Robot is a Multi-functional kit. It can implement advanced features like object following, obstacle avoidance and so on. It can also be remote controlled by an Android Smartphone's APP and IR remote control. This guide describes in detail how users can make the robot work.

1.1 Package Contents

Figure 1-1 shows the package contents of the Self-Balancing Robot kit



Figure 1-1 Self-Balancing Robot package

The Self-Balancing Robot kit package contents:

① Self-Balancing Robot

② Lithium Battery

③ Lithium Battery Charger

④ IR Remote Control

⑤ Mini USB Cable

⑥ Micro USB Cable

⑦ Quick Start Guide

Chapter 2

Components and Functions

2.1 Parts and Functions

This chapter will describe the robot's components and the functions of the components, such as the ultrasonic module, motors, wheels and driving board power interface. **Figure 2-1**, **Figure 2-2**, **Figure 2-3** and **Figure 2-4** show all the components and specify their function.

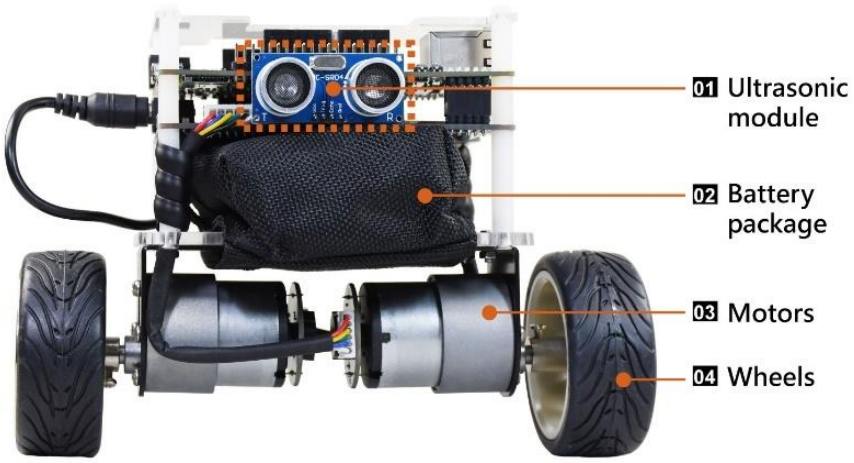


Figure 2-1 Self-Balancing Robot components

(1) Ultrasonic module：implements obstacle avoidance.

(2) Battery package：helps to protect the batteries and avoid bumping and damaging the battery.

(3) Motors：drive the Self-Balancing robot wheels.

(4) Wheels：implements the Self-Balancing robot's movement.

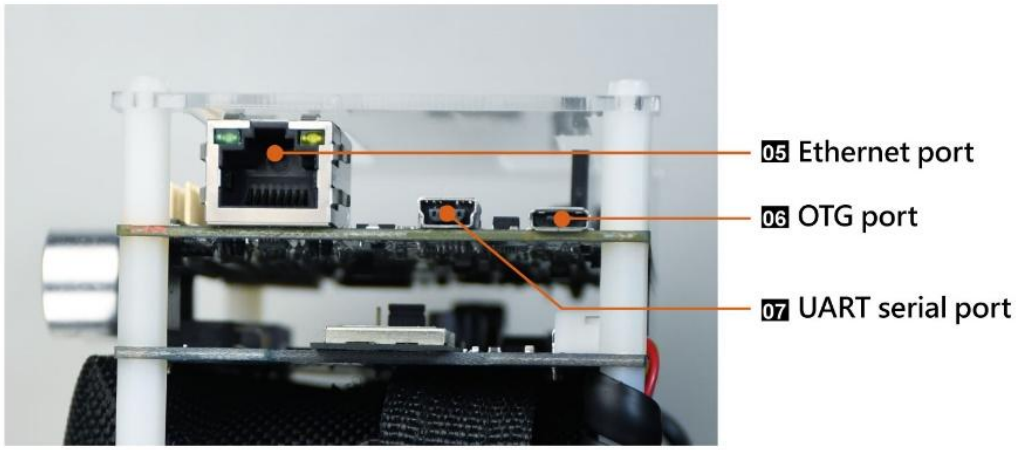


Figure 2-2 Self-Balancing Robot components

(5) Ethernet port：implements connecting to the ethernet when users develop their own designs on the Self-Balancing robot.

(6) OTG port：implements Host or Device mode when users develop their own design on the Self-Balancing robot.

(7) UART serial port：implement the communication between the board and PC when users develop their own design on the Self-Balancing robot.

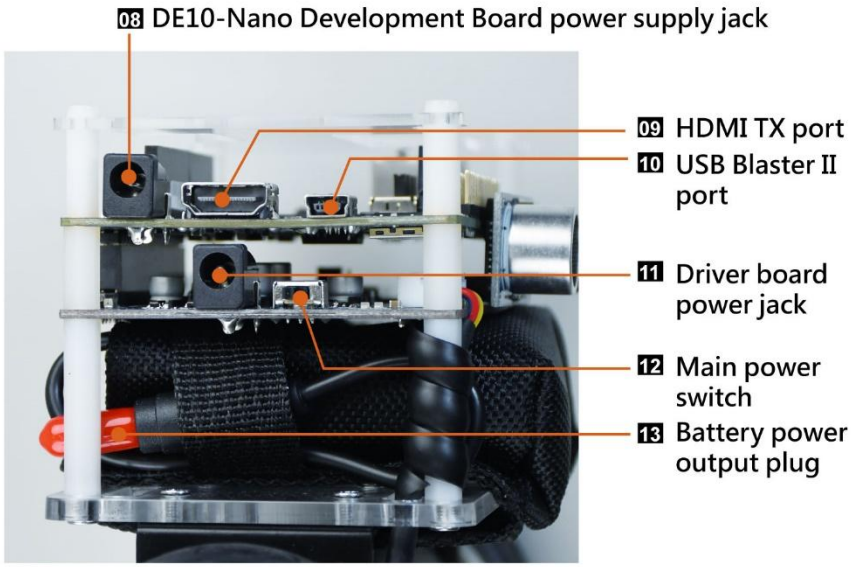


Figure 2-3 Self-Balancing Robot components

(8) DE10-Nano Development Board power supply jack：5V power supply port.

(9) HDMI TX port：users can connect the Displayer to HDMI interface when they make image processing designs.

(10) USB Blaster II port：users can download their own programs to the board through the connector.

(11) Motor driver board power jack：connects to the battery power supply port and provides power for the Self-Balancing robot.

(12) Main power Switch：power on or power off the Self-Balancing robot.

(13) Battery power output plug：connect to motor driver board power port.

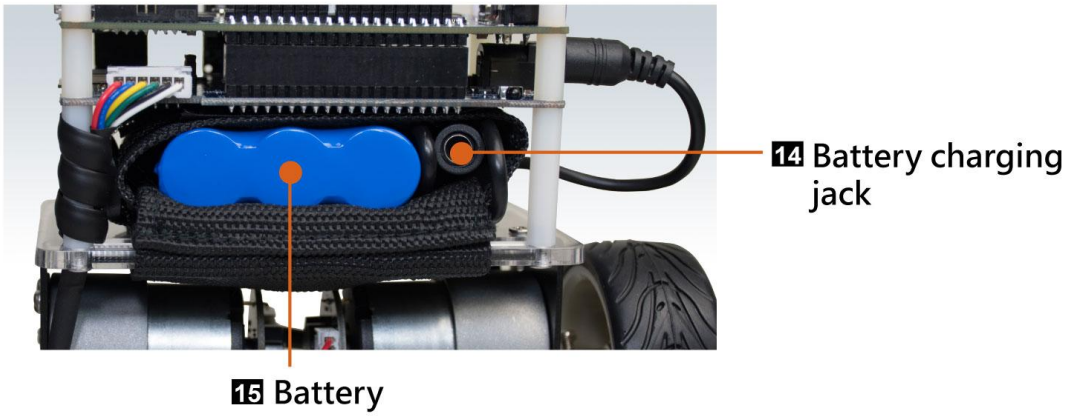


Figure 2-4 Self-Balancing robot components

(14) Battery charging jack：connects the charger to charge the battery.

(15) Battery：the Self-Balancing robot's power source.

2.2 DE10-Nano Kit and Motor Driver Board

The Self-Balancing Robot control system consist two boards, Terasic DE10-Nano FPGA board and Motor Driver board (BAL board) as shown in **Figure 2-5**. The FPGA on the DE10-Nano is responsible for all the functions of the balance and control system. The motor driver board receives the control signal from the de10-nano to control the motor rotation. There are also some sensors and communication devices on the motor driver board. These devices can provide the status data of the robot and external communication interface to FPGA. For detailed hardware information, please refer to 03_Hardware_Manual.pdf which can be found in the CD package.

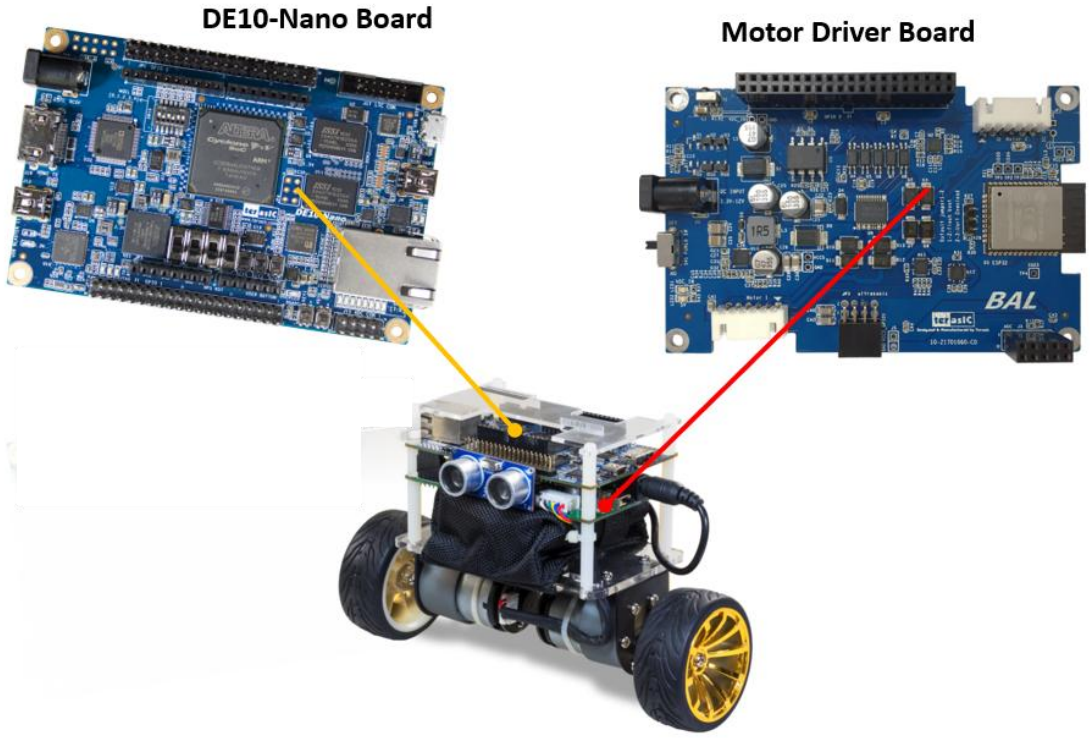


Figure 2-5 DE10-Nano and Motor Driver Board

Chapter 3

Setup Elements

This chapter will introduce the switches and buttons that can be set on the Self-Balancing Robot. It explains the meaning and function of the setting.

3.1 Configuration Mode Switches

The Self-Balancing Robot equips a Cyclone SoC FPGA, which means that the ARM processor is embedded in the FPGA. Therefore, there are two processor options available to control the Robot. One is to use the ARM processor and the other is implement a NIOS II processor in the FPGA. The kit provides the factory code for both processor. To switch between these two processor modes, user need to select via the mode select switch(MSEL[4:0]).

If user want to select ARM to control the robot, the MSEL[4:0] needs to switch to "01010", as shown in **Figure 3-1**. Thus, when the robot is power on, the FPGA will boot from the Micro SD card and run the Linux by ARM processor to control the robot.

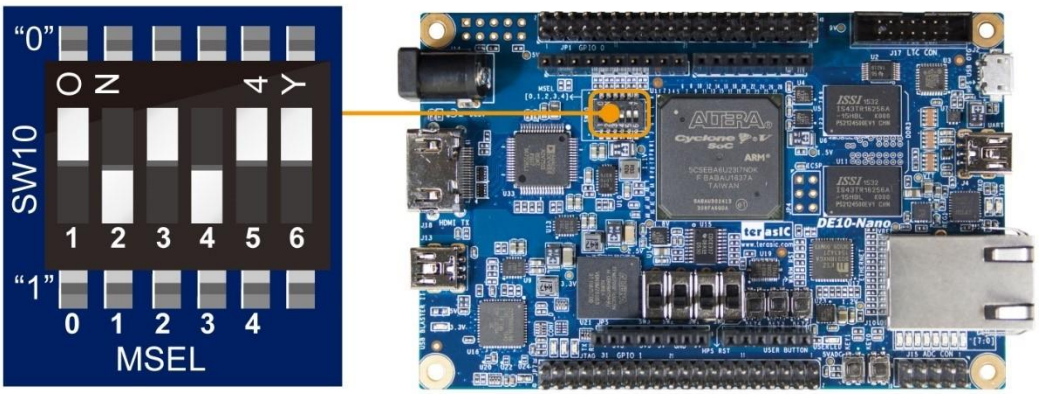


Figure 3-1 Set MSEL[4:0] to 01010

When MSEL[4:0] is set to "10010", as shown in **Figure 3-2**.The FPGA will boot from the configuration device(EPCS). Then, after FPGA is configured, the NIOS II processor will control the robot. For users who is beginners to learn FPGA, using NIOS II processor will be easier than deal with the ARM processor. The factory default setting of the robot is Nios version, when switch the robot to ARM version and power on it, users need to wait about 20 seconds before the LED7 is light on.

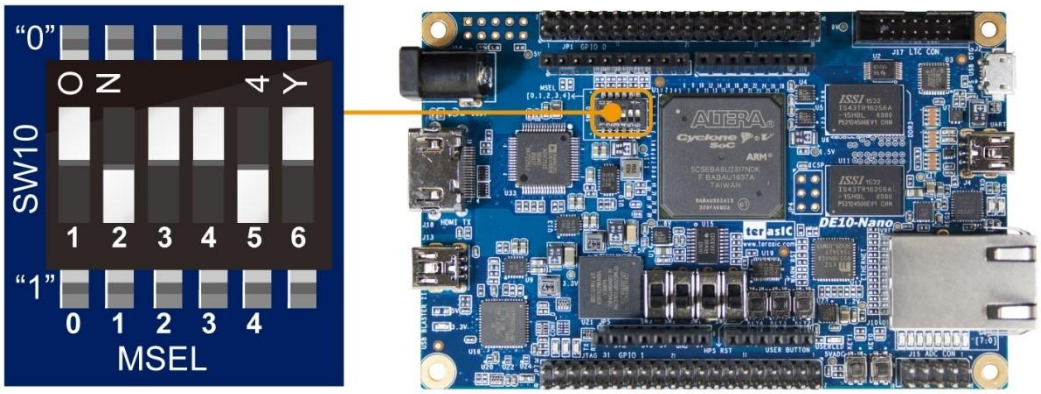


Figure 3-2 Set MSEL[4:0] to 10010

3.2 Operation Mode Switches

Figure 3-3 shows the SW0 and SW1 on the DE10-Nano board, **Table 3-1** describes the corresponding modes and functions when SW0 and SW1 are set to different positions.

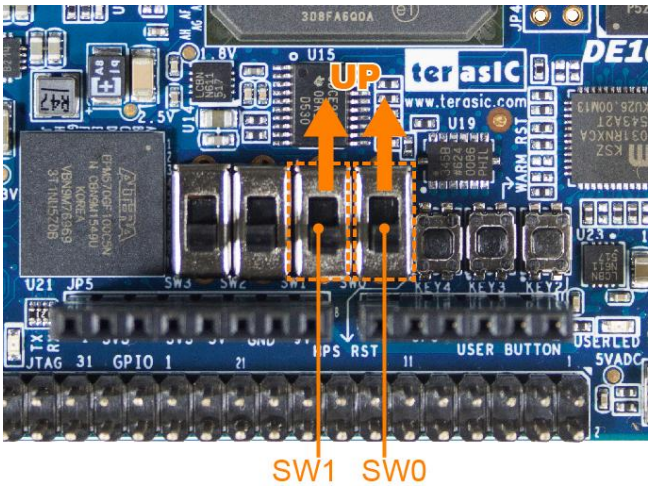


Figure 3-3 SW0 and SW1 on DE10-Nano board

Table 3-1 SW0 and SW1 purpose

SW[1:0] Setting	Robot mode and function	Description
00	Default mode (Bluetooth and IR mode)	The robot can be controlled by smartphone APP and IR remote control
10	Default mode and Obstacle Avoidance	The robot can be controlled by a smartphone's APP and IR remote control, it implements the obstacle avoidance function (Only IR remote control supports)
01	Object following & Obstacle Avoidance	The robot implements the object following and Obstacle Avoidance (In this mode, the robot will not be controlled by smartphone APP and IR remote control)
11	Debug mode	Only supports ARM version robot, the control program will stop running, users need to reboot the robot or run the program again to control the robot. Normally it is used to debug the robot.

3.3 LEDs on the Motor Driver Board

Figure 3-4 shows the LED1 and LED2 on the motor driver board, **Table 3-2** describes the functions of LED1 and LED2.

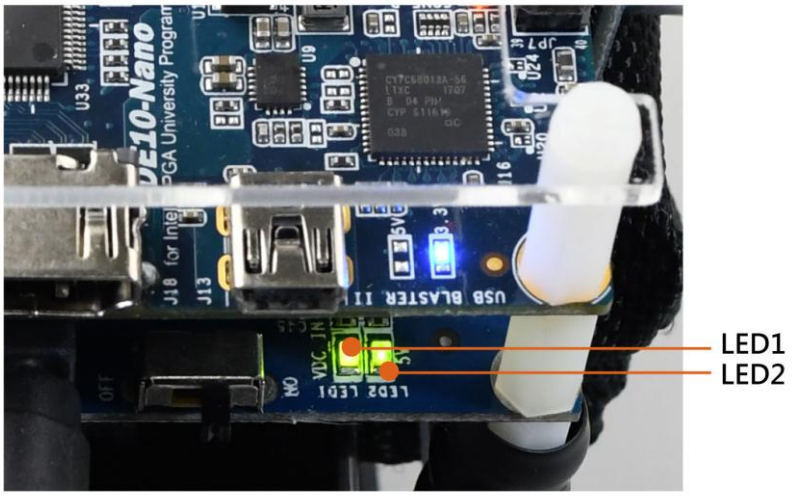


Figure 3-4 LED1 and LED2 on the motor driver board

Table 3-2 Motor driver board LEDs functions

LED name	Description
LED1	Indicates the power supply status to motor driver board
LED2	Indicates the motor driver board provides 5V power to the DE10-Nano board

3.4 LEDs on DE10-Nano Board

Figure 3-5 shows the 3.3V power LED, CONF_D LED and other LEDS on the DE10-Nano board, **Table 3-3** describes the LEDs functions.

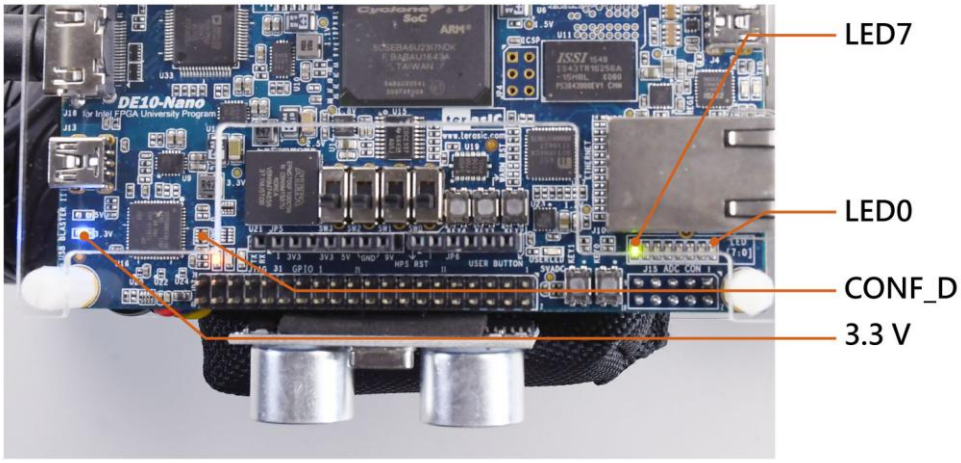


Figure 3-5 Indicator LEDs on DE10-Nano board

Table 3-3 Indicator LEDs on the DE10-Nano board

LED name	LED status	Description
3.3V power LED	Light On	Illuminate when 3.3V power is active
CONF_D	Light On	DE10-Nano board Configuration done
LED7	Light On	Robot is keeping balanced status
LED6~5	1--Light On 0--Light off	00--robot is in default mode (Bluetooth & IR control) 01--robot is in default mode and implements obstacle avoidance function 10--robot implements object following and obstacle avoidance function
LED4	Light On	Battery power supply voltage is lower than 10.5V
LED3	Light On	Robot is turning right
LED2	Light On	Robot is turning left
LED1	Light On	Robot is moving backward
LED0	Light On	Robot is moving forward

Note: When LED3~0 are all lit indicates the robot is in DEMO mode.

Chapter 4

Basic Operations

This chapter illustrates the basic operations on how to start the robot when users receive it.

4.1 Connect Power Port

Remove the red protective cap from the output plug of the battery power, as shown in **Figure 4-1**.

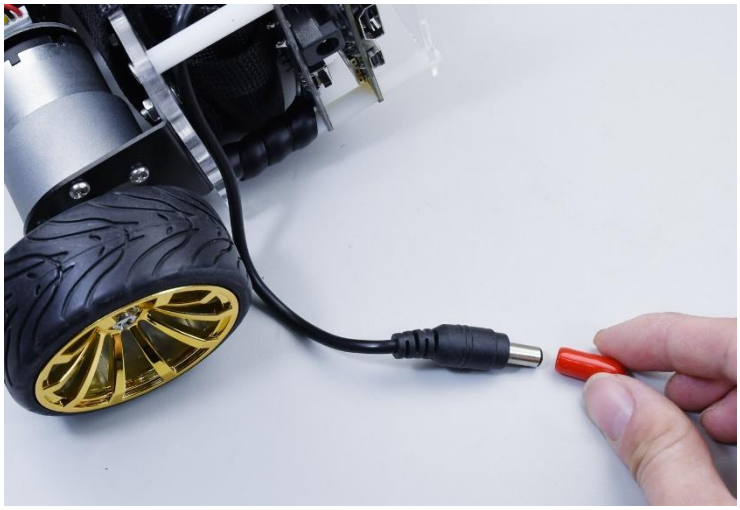


Figure 4-1 Remove the protective cap

Insert the output plug of battery power into the input port of the motor driver board, as shown in **Figure 4-2**.

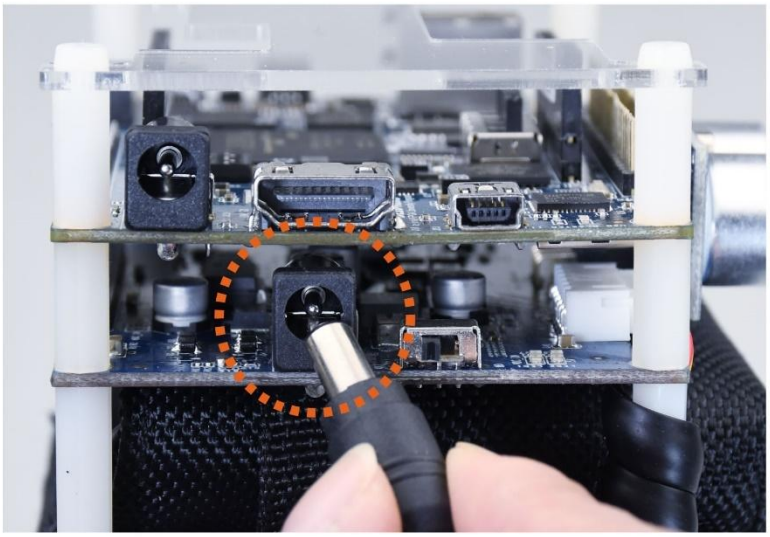


Figure 4-2 Connect the battery power connector to the power input port

4.2 Power on the Robot

Place the robot on the plane, keep it in a horizontal state, then set the power switch of the motor driver board to ON position, as shown in **Figure 4-3**.

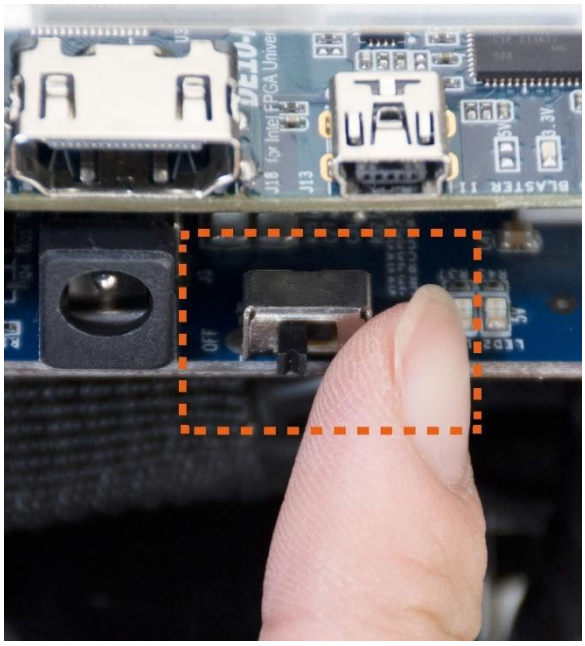


Figure 4-3 Set SW1 to ON position

4.3 Keep a Balanced State

When the LED7 on the DE10-Nano board is lit, please release the robot, it will keep balance automatically, as shown in **Figure 4-4**.

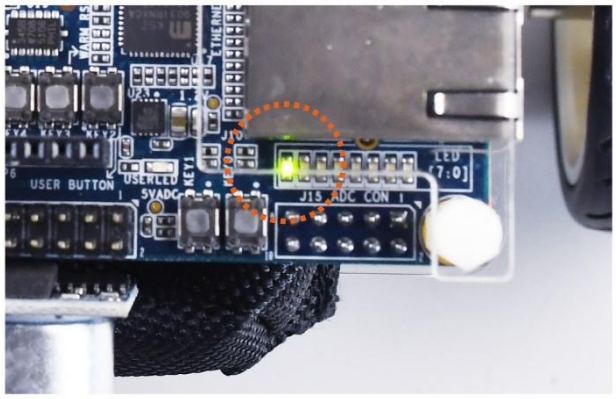


Figure 4-4 LED7 lights on shows the robot is in balance state

4.4 Attitude Control

The robot can perform posture recognition in real time through the acceleration sensor and the gyroscope and achieve balance by controlling the motors and adjusting the posture. For example, pick up the robot, it will detect that its current state is not in a horizontal state, when the robot is placed on a horizontal plane, the robot will access its current balance status and adjust accordingly and keeping it in balanced state without any external help. If an external force causes the robot to tilt forward, the motors will quickly produce the forward motion torque to compensate for the angle of the tilt and maintain the balance of the robot. If an irregular object is placed on the robot, the robot body will maintain balance.

Chapter 5

Advanced Features Demonstration

Based on the DE10-Nano SoC FPGA platform, Terasic's Self-Balancing Robot can implement attitude algorithm, perform motion control, and execute movements autonomously, such as moving forward, turning right & left, power monitoring, object following and obstacle avoidance. Line following and obstacle avoidance are described below.

5.1 Object following and Obstacle avoidance demonstrate

Once the ultrasonic module is assembled on the robot, object following and obstacle avoidance can be implemented with the module.

When the robot is in object following and obstacle avoidance is on (SW[1:0] is on "01" position, as shown in **Table 3-1**), if the ultrasonic sensor detects the obstacle is in front of the robot and the distance is within 10 cm, the robot will automatically move backward, which will implement the obstacle avoidance function, as shown in **Figure 5-1**.

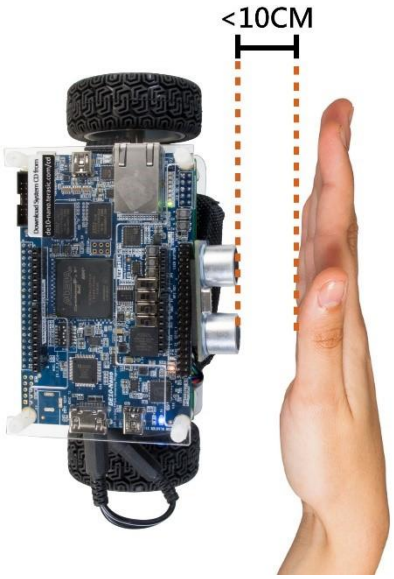


Figure 5-1 Obstacle avoidance demonstrate

When an object is in front of the ultrasonic module and moves slowly and the distance is within 10 cm~20 cm, the robot will continue to move along with the object, which will implement the object following function, as shown in **Figure 5-2**.

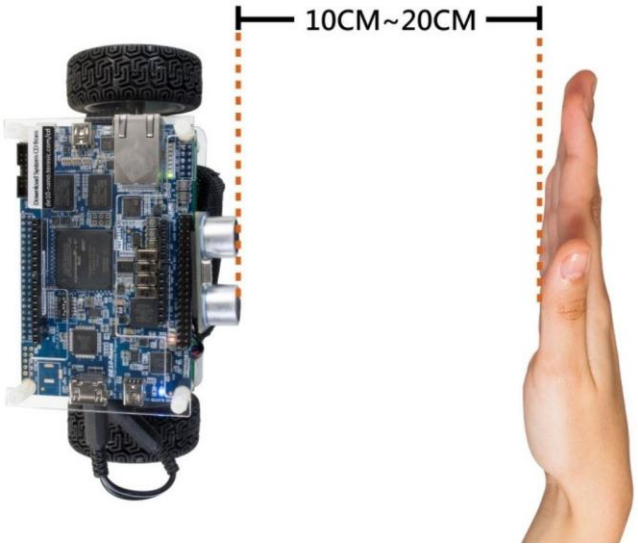


Figure 5-2 Object following demonstrate

5.2 Smartphone APP Control

The robot can be remote controlled by a Smartphone APP, this section describes how to control the robot by APP.

5.2.1 Android APP Control

■ Download APP

Users can download and install the Android Smartphone APP by scanning the QR code below (See **Figure 5-3**) or download the APP file from the link:

http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=238&No=1096&PartNo=4

After the installation is completed, the APP icon is shown as **Figure 5-4**,

Figure 5-3 Download QR code of the APP



Figure 5-4 Android APP icon

■ Connect APP and Robot

Power on the robot, set SW0~3 of the DE10-Nano to Down position, as shown in **Figure 5-5**.

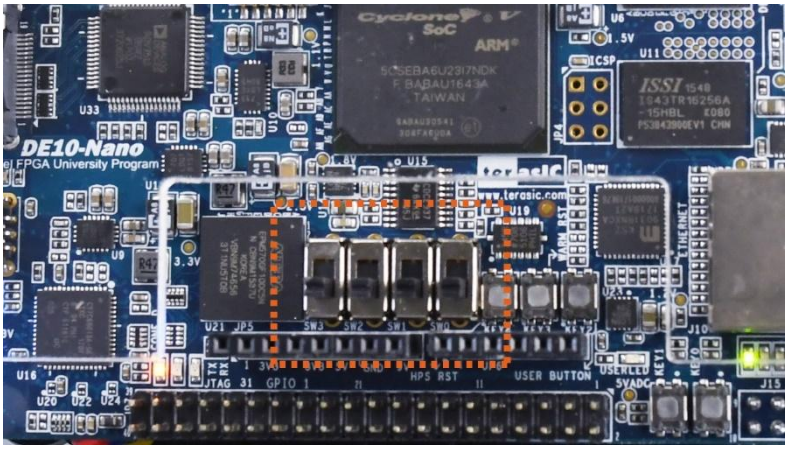


Figure 5-5 Set SW0~3 to Down position

Run the robot APP, click the search icon on the upper right corner of the APP GUI, as shown in **Figure 5-6**

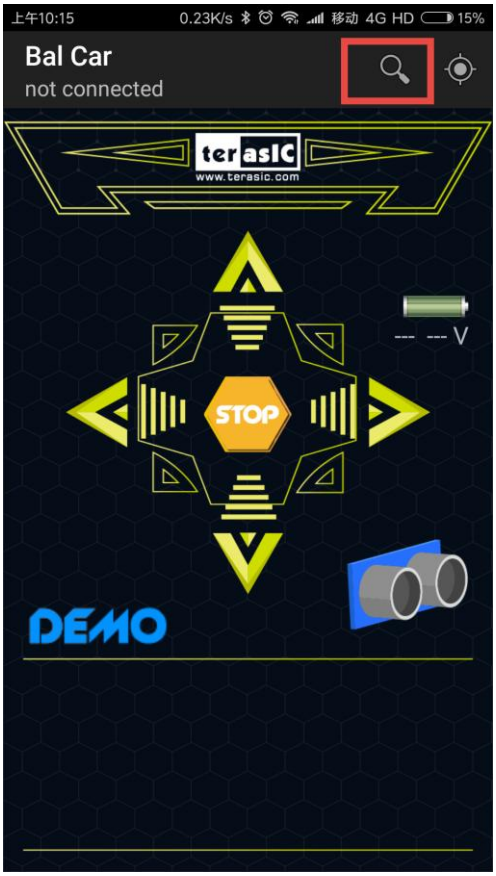


Figure 5-6 Click the search icon on the APP GUI

When the actual robot device name appears, select it as the device to connect, as shown in **Figure 5-7.**

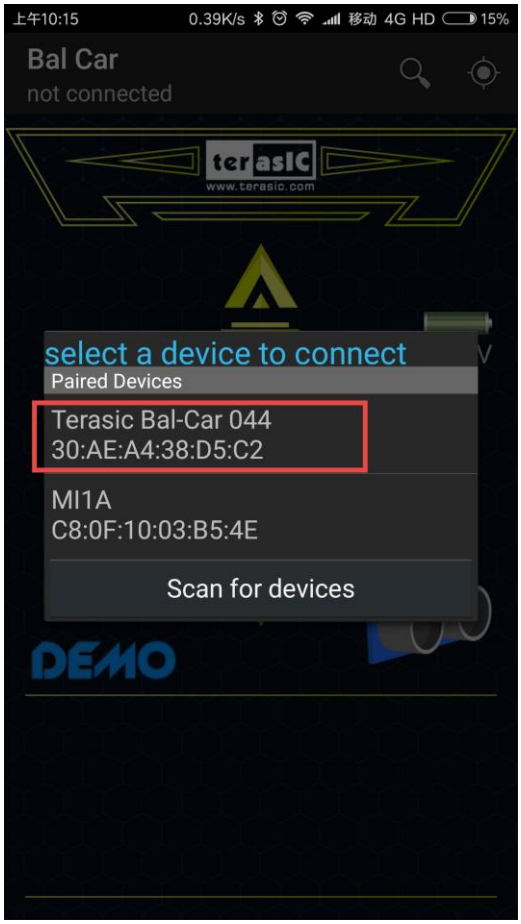


Figure 5-7 Connect to robot Bluetooth device

After connecting to the robot successfully, it will display connection status "connected to Terasic Bal-Car 0XX" at the top left corner of the APP, as shown in **Figure 5-8**.

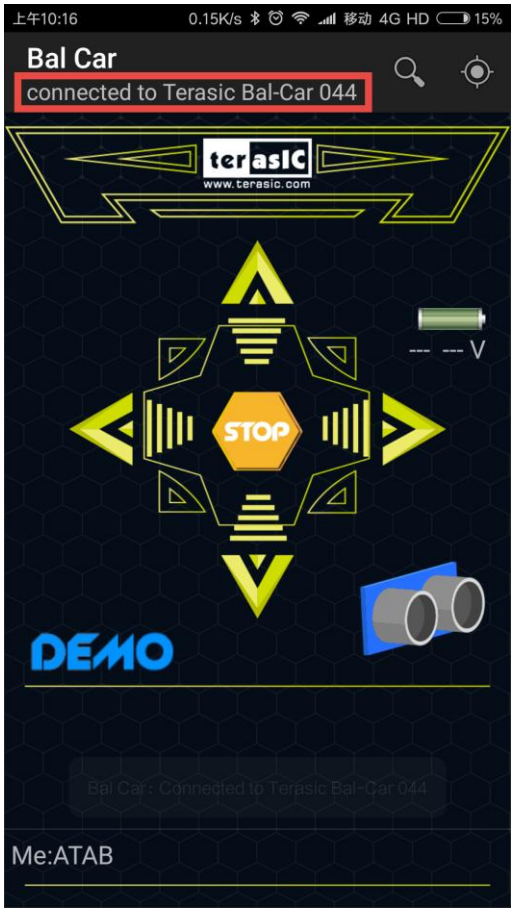


Figure 5-8 Connected to the robot successfully

■ Control the Robot

Now users can click the yellow direction keys, STOP key and DEMO key to control the robot, **Figure 5-9** shows the direction keys functions.

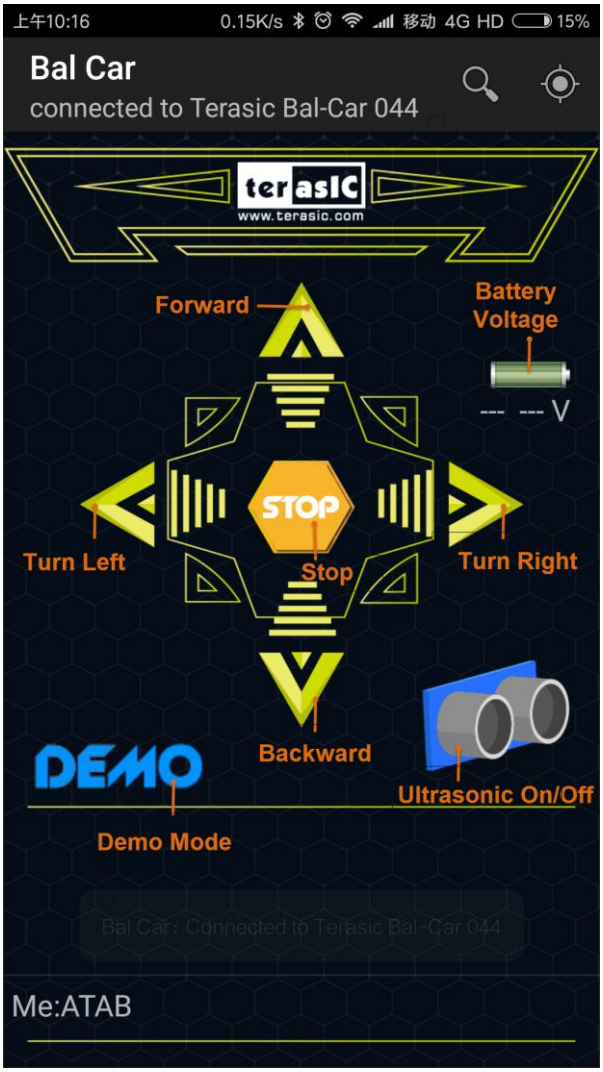


Figure 5-9 APP GUI

- Forward：robot moves forward (Ensure the ultrasonic module is installed)
- Backward：robot moves backward
- Turn left：robot turns left
- Turn right：robot turns right
- STOP：robot stops moving
- Ultrasonic On/Off: Enable/Disable ultrasonic module for **obstacle avoidance** function
- DEMO mode：when clicking the DEMO button, the robot will perform the scheduled action. After finishing the action, it will exit the DEMO mode automatically; if you click DEMO button again or click Stop button during DEMO mode, the robot will exit the DEMO mode automatically and keep balance where it is; if you click Forward/Backward/Left/Right button during the DEMO mode, the robot will exit the

DEMO mode immediately, and perform the action according to the button

5.3 IR Remote Control

Figure 5-10 shows the remote control for the robot, point IR remote control to the robot, the robot will move forward when users press key 2, the robot will stop moving when user press key 5. The key 8 is used to move the robot backward, key 4 is used to turn the robot to the left, and key 6 is used to turn the robot to the right, Key A is used to perform the scheduled action. Table 5-1 shows the functions of each key number on the controls.

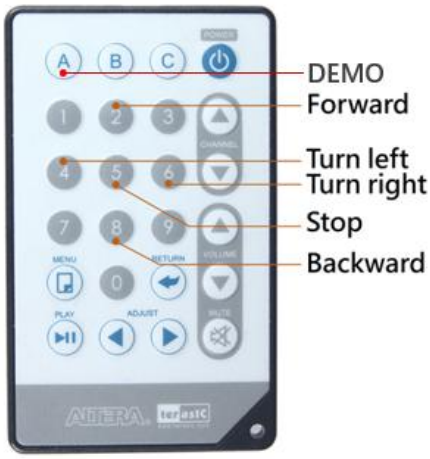


Figure 5-10 Robot remote control

Table 5-1 Remote control function keys

Key numbers	Function
2	Forward
5	Stop
8	Backward
4	Turn left
6	Turn right
A	DEMO

Chapter 6

Charging the battery

The robot is powered by a three-section of lithium battery (the specific parameters of the battery can be seen outside the battery package). When the power is less than 10.5V, the LED4 on DE10-Nano board will light up, indicating that the battery needs to be charged in time; and the APP will also show the battery power level. If the lithium battery starts charging after it is completely drained and is completely unable to supply power to the robot, it will take up to 2 hours the battery to be fully charged. The battery charging steps are as follows:

Power off the robot, pull out the power cable, and take the battery out of the robot's onboard battery storage space, as shown in **Figure 6-1**.



Figure 6-1 Robot battery

As shown in **Figure 6-2**, connect the charger to battery connector.



Figure 6-2 Connect charger to battery connector

As shown in **Figure 6-3**, plug the charger into the AC 220V or 110V power outlets, after the power is fully charged, the LED on the charger will light up green, then unplug the charger.



Figure 6-3 Charge the battery

Chapter 7

Restore Factory Setting

This chapter will introduce the switches and buttons that can be set on the Self-Balancing Robot. It explains the meaning and function of the setting. There are two versions of factory code: ARM and NIOS CPU-controlled version. So that, the following will describe the two methods of restoring.

7.1 ARM Version Restoring

The factory code of AMR version is stored in the Micro SD Card. The following will describe how to restore factory code in the Micro SD Card.

- **Required Equipment:**
  - PC: Write Linux image file into SD card
  - Micro SD Card: 8GB minimum
  - Micro SD Card reader: Write the SD Micro SD card

- **Software and file requirements:**
  - Win32DiskImager.zip: the tool which is used to write image file to Micro SD Card, it's located at CD\Tool\
  - *de10_nano_balance_car.zip*: the compressed demo image file for ARM version robot, it's located at http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=1096&PartNo=4

- **Steps:**

- Copy the *de10_nano_balance_car.zip* to PC and unzip it to get *balance_car.img* file.
- Insert the Micro SD card into SD card reader and connect the card reader to PC USB port.
- Copy *Win32DiskImager.zip* to PC and unzip it, execute *Win32DiskImager.exe* in the unzipped Win32DiskImager folder.
- As shown in **Figure 7-1**, choose *de10_nano_balance_car.img* for Image File.

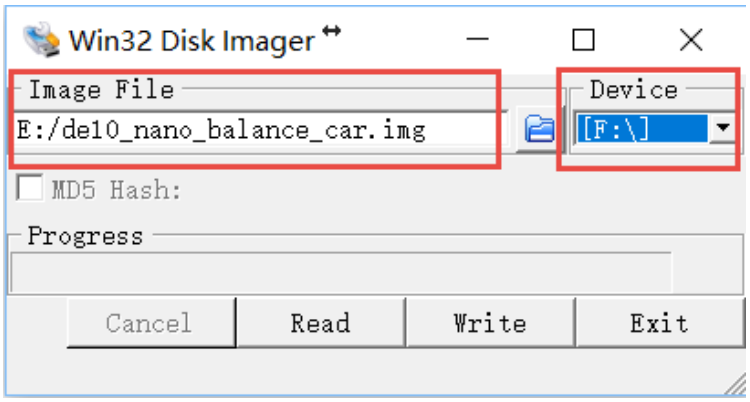


Figure 7-1 Win32DiskImager window

- Choose the drive disk of Micro SD card for Device.
- Click "write" to start writing the image file to the microSD card. Wait until the image is written successfully.
- As shown in **Figure 7-2**, insert the Micro SD card into the robot. And set the mode switch(SW10) MSEL[4:0] to "01010", as shown in **Figure 7-3**.

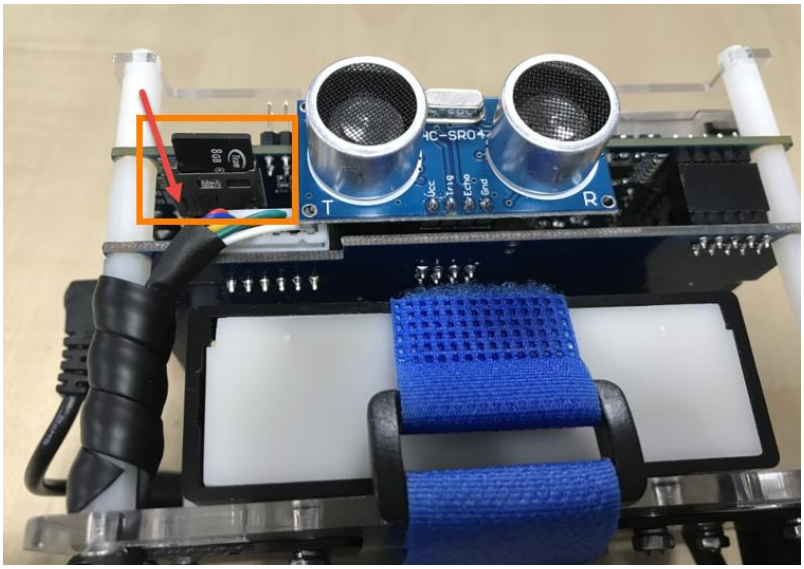


Figure 7-2 Insert the Micro SD card into the robot

- Power on the robot then start using it.

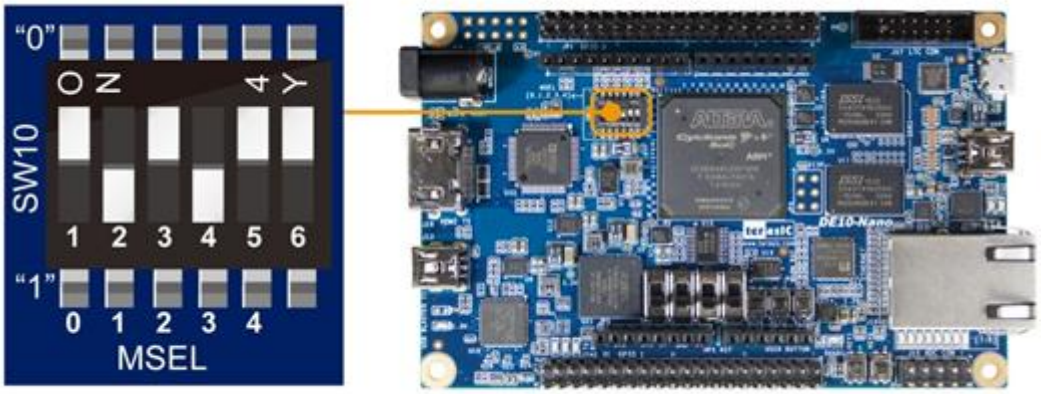


Figure 7-3 Set MSEL[4:0] to "01010" for ARM Version Mode

7.2 NIOS Version Restoring

The factory code of NIOS version is stored in the EPCS device. The following will describe how to restore factory code in the EPCS.

- **Required Equipment:**
  - PC: configuring the jic file to EPCS device on the robot
  - Mini USB Cable x 1: Connect robot to PC for configuring the file

- **Software and file requirements:**
  - Ensure the Intel Quartus tools is installed properly.
  - The compressed .jic file for Nios version robot, it's located at CD\Demonstration\factory\nios\

- **Steps:**
  - As shown in **Figure 7-4**, Connect a USB cable to the USB Blaster II connector on the robot and the PC.

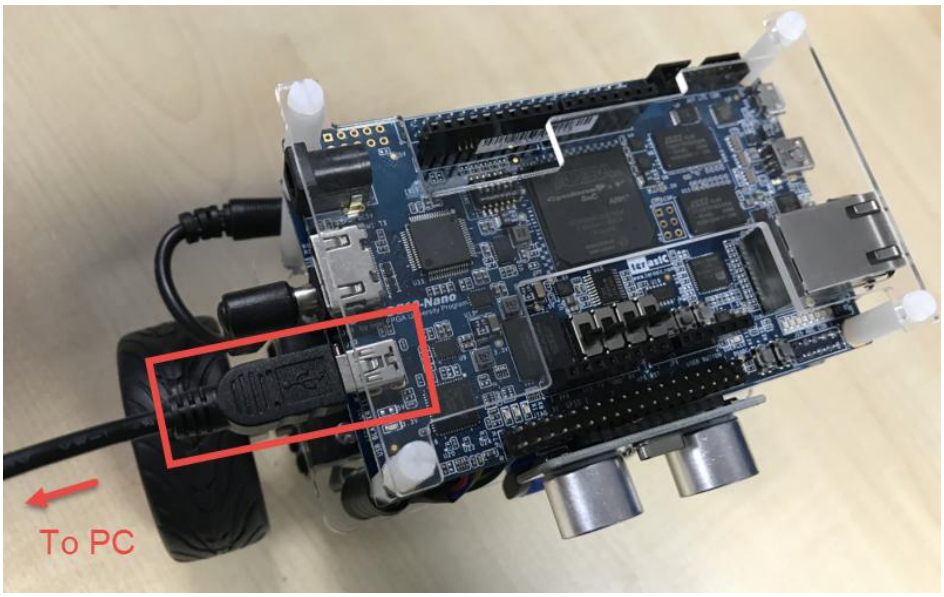


Figure 7-4 Connect the robot to PC via USB Mini Cable

- As shown in **Figure 7-5**, set the mode switch (SW10) MSEL[4:0] to "10010".

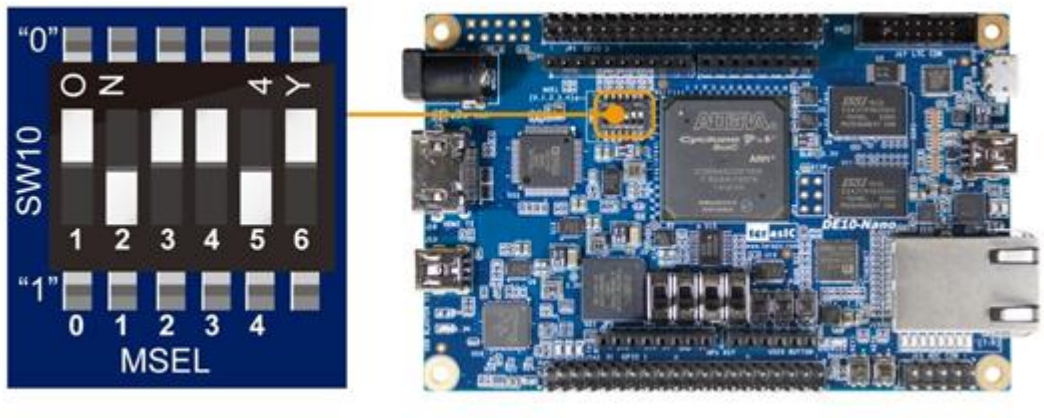


Figure 7-5 Set MSEL[4:0] to "10010" for NIOS Version Mode

- Copy demo_batch_jic.zip to PC and unzip it to get demo_batch_jic folder.
- As shown in **Figure 7-6**, input number 3 in the pop-up command window and click Enter key, it will start to configure the .jic file to EPCS device.
- After the configuration is completed, remove the USB cable. Power on the robot and verify if the code is written right.

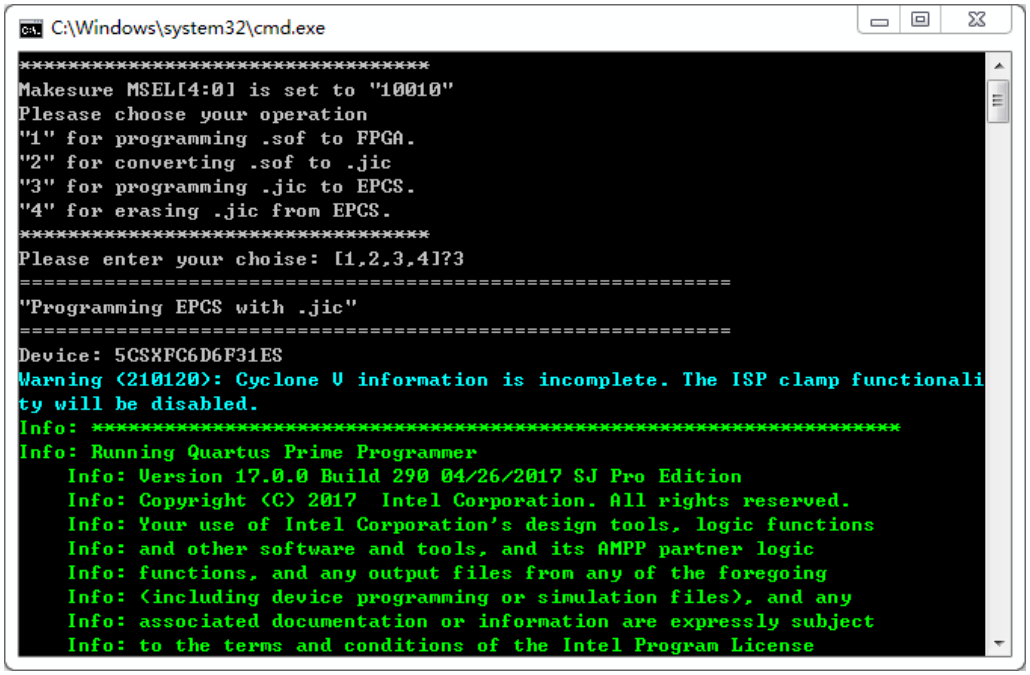


Figure 7-6 Command shell for write .jic file into EPCS device

Additional Information

Getting Help

Here is the contact information where you can get help if you encounter problems:

- Terasic Technologies
  9F, No.176, Sec.2, Gongdao 5th Rd, East Dist, Hsinchu City, Taiwan 300-70
  Email : support@terasic.com
  Web : www.terasic.com

Revision History

Date	Version	Changes
2018.03.16	First publication	
2018.05.29	V1.1	Modify Table 3-1 and section 5.1 description
2018.07.11	V1.2	Modify description in section 3.4 and 5.1
2020.10.27	V1.3	Delete iOS app descriptions