

ARM

Design and Implementation of ARM Intelligent Vehicle Control System

1501

	I
Abstract	II
1	1
1.1	1
1.2	1
1.3	3
1.4	3
2	4
2.1	4
2.2	4
2.3	4
3	5
3.1	5
3.2	5
3.3	6
4	7
4.1	7
4.2	12
5	28
5.1	28
5.2	30
5.3	30
6	35
	36
	38

			ARM
	PID		ARM
ARM Cortex-M3		LM3S615	1024
	IRM8601S		ARM
ARM		PID	

Abstract

With the continuous development of the automotive industry, with the support of science and technology, automotive intelligence has become a trend. Traditional automobile driving is mainly manual operation, and drivers are sometimes disturbed by other factors, so there are certain safety risks. The design and use of intelligent vehicles can not only help people save resources, but also solve the work that people cannot complete. This paper combines ARM embedded chip, designs and implements intelligent vehicle control system, aiming at continuously promoting the development of automobile intelligence.

Based on the incremental PID control algorithm and graph theory algorithm, the ARM intelligent car control system is designed and implemented. With ARM Cortex-M3 microcontroller LM3S615 as the core control chip, the speed of the car is output by an integrated 1024-line encoder, and the speed and direction parameters are fed back to the DC motor and gyroscope. The wall information is detected and fed back by the integrated infrared sensor IRM8601S, and the contour map is used to realize the automatic driving, obstacle avoidance and road search of the ARM intelligent car. The best choice is the best. Optimal path and other functions.

Key Words: ARM Embedded Incremental PID Control Intelligent Vehicle
Graph Theory Closed-loop Control

1

ARM

ARM

[1]

1 1

ARM

ARM

1 2

50

1954

1986

1995

1996

-

Navlab

21

European Land Robot Trial

DARPA

1 1992

2 2003

130 / ~170 /

3 2006

Cyber Car

4 2014

10

863

[2]

5 2018

1 3

1 4

ARM

2

2 1

ARM

1 ARM

2 ARM

3 ARM

2 2

ARM

1

2

3

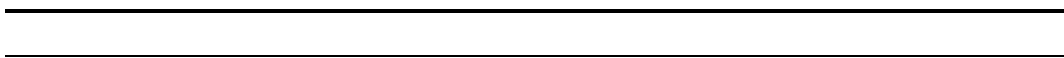
4

5

6

2 3

1



ARM

2

v

θ

5m/s

7.4v/350mAH

96mm

84mm

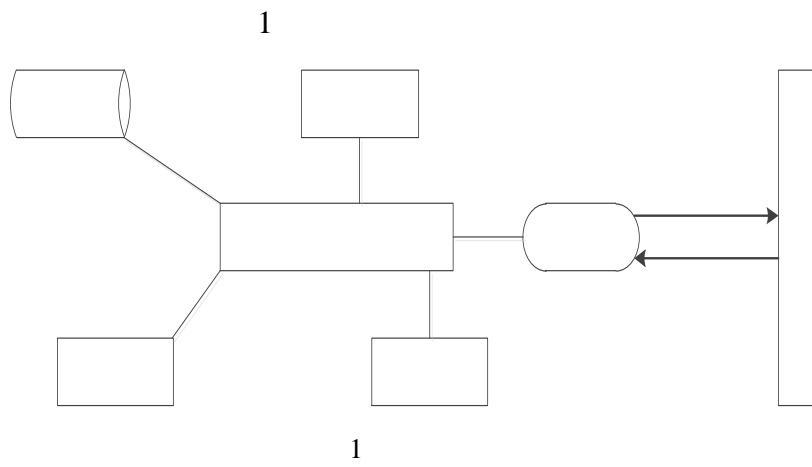
300g



3

ARM

3 1



3 2

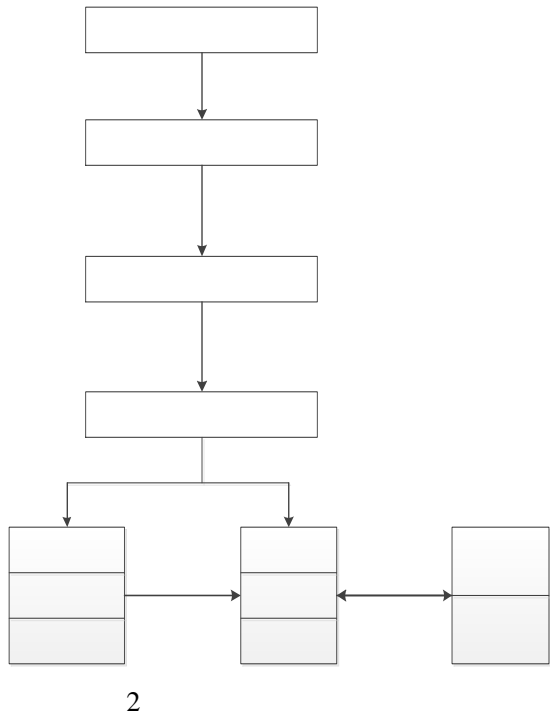
/

2

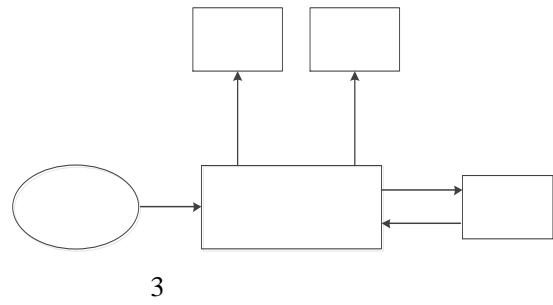
/

3

5



2

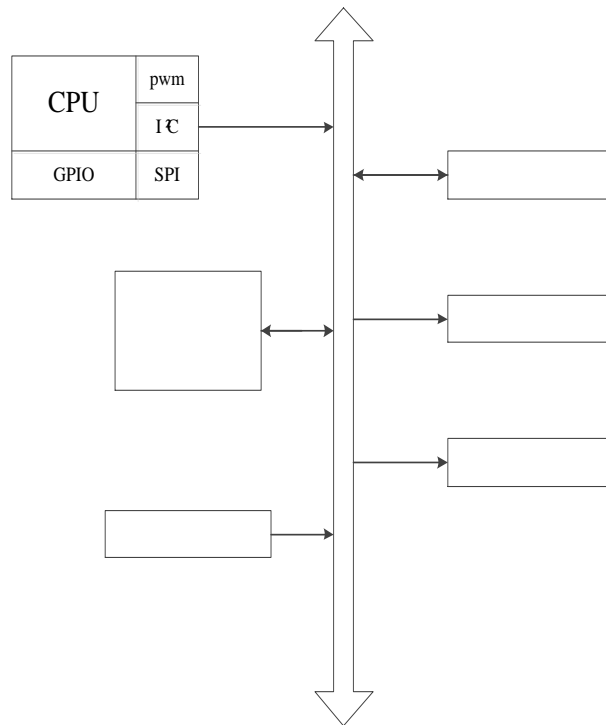


3

3 3

CPU

4



4

4

4 1

4 1 1

LM3S615

ARM®

Cortex™-M3

8

16

32

[3-6]

4 1 2

1717SR

7.4V

1.96W

14000rpm^[7-8]

1

PWM

5

PWM

V1

U_i

PWM

V1

U_s

t₁

U_i

V1

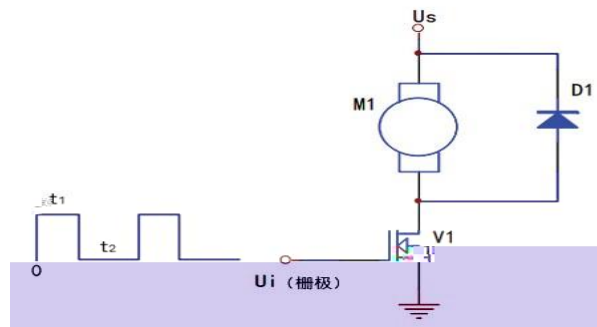
t₂

U_o

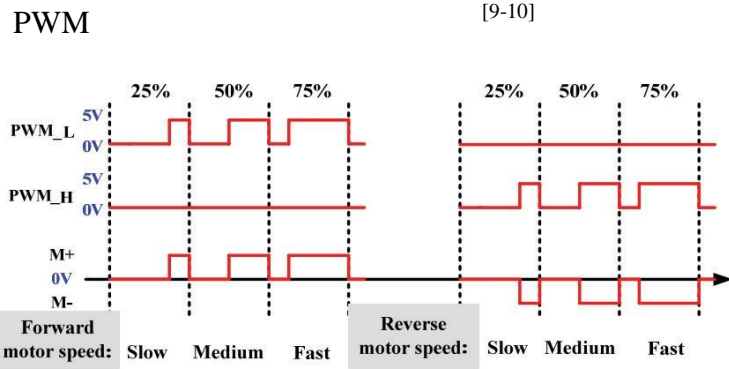
1

$$U_o = (t_1 * U_s) / (t_1 + t_2) = U_s * (t_1 / T) = \alpha * U_s \quad (1)$$

α

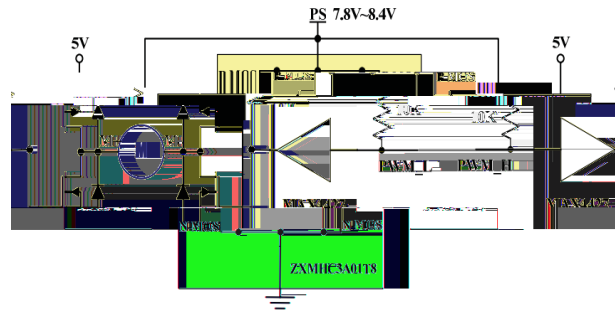


5 PWM



6 PWM

2



7

7

PWM PWM_H L PWM_L H

PWM_H H PWM_L L

PWM_H L PWM_L L

PWM_H H PWM_L H

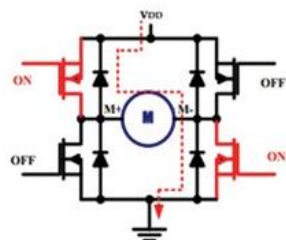
PWM

2

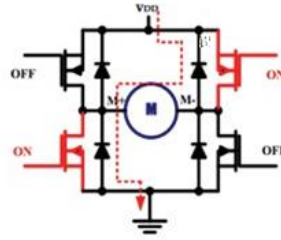
8

2

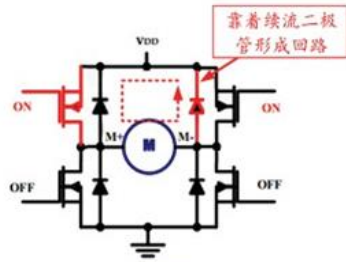
PWM_H	PWM_L
L	L
L	H
H	L
H	H



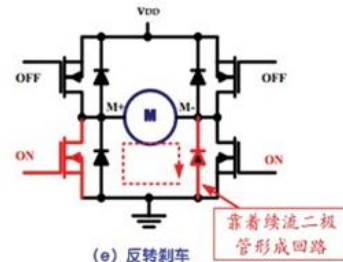
(b) 正转



(d) 反转



(c) 正转刹车



(e) 反转刹车

8

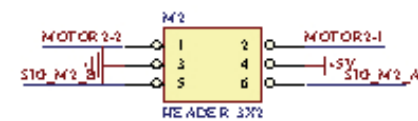
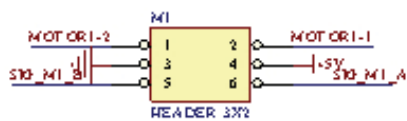
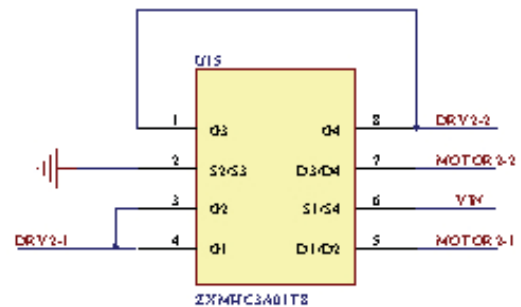
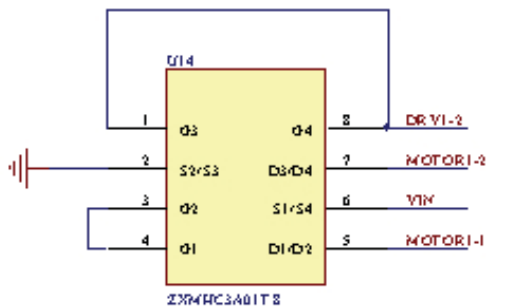
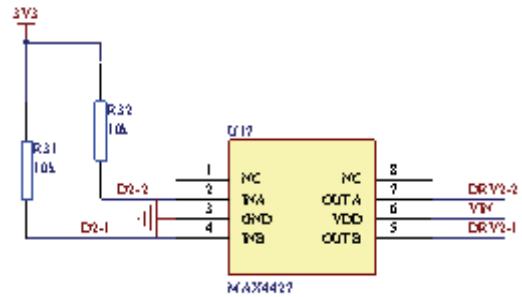
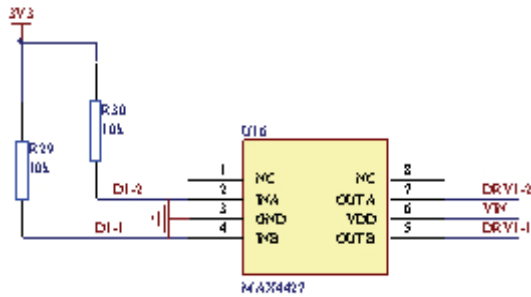
3

9

MAX4427

HEADER

ZXMHC3A01T8 H



9

4 1 3

ARM

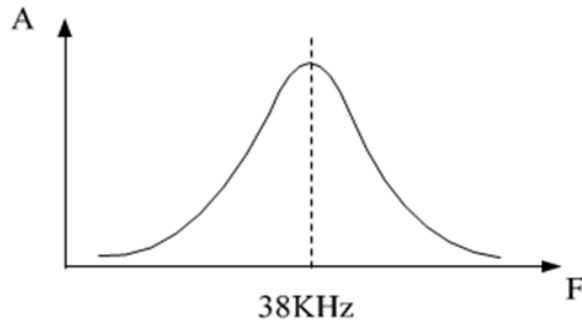
6

38KHz

[11-13]

10

38KHz



10

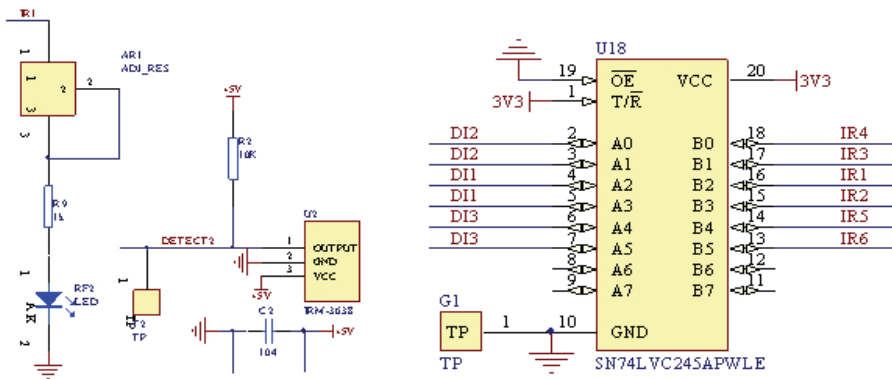
ARM

11

IR1

SN74

4.2



11

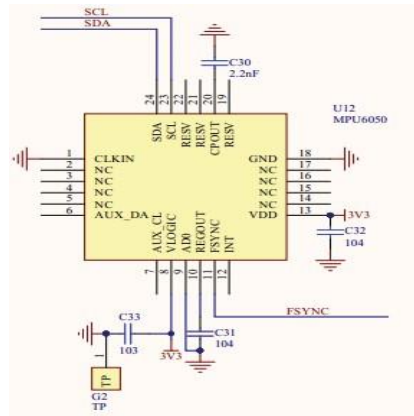
4 1 4

MPU6050

MPU6050

16 ADC^[14-16]

12



12 MPU6050

4 1 5

4 2

4 2 1

1 PID

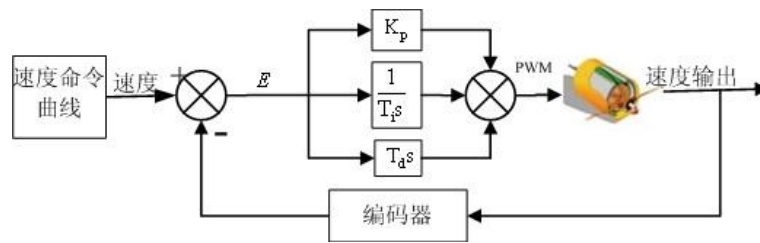
PWM

PWM

PI

PI

14



14 PI

14

e PI PWM

DC

e 0

PID

ΔU_k

ΔU_k

PID

2

[22]

2

k-1

3

$$u_k = k_p \left[e_k + \frac{T}{T_i} \sum_{j=0}^k e_j + Td \frac{e_k - e_{k-1}}{T} \right] \quad (2)$$

$$u_{k-1} = k_p \left[e_{k-1} + \frac{T}{T_i} \sum_{j=0}^{k-1} e_j + Td \frac{e_{k-1} - e_{k-2}}{T} \right] \quad (3)$$

2

3

PID

4

$$\begin{aligned} \Delta u_k = u_k - u_{k-1} &= k_p(e_k - e_{k-1} + \frac{T}{T_i}e_k + Td \frac{e_k - 2e_{k-1} + e_{k-2}}{T}) \\ &= k_p(1 + \frac{T}{T_i} + \frac{Td}{T})e_k - k_p(1 + \frac{2Td}{T})e_{k-1} + k_p \frac{Td}{T}e_{k-2} \\ &= Ae_k + Be_{k-1} + Ce_{k-2} \end{aligned} \tag{4}$$

$$A = k_p(1 + \frac{T}{T_i} + \frac{Td}{T})$$

$$B = k_p(1 + \frac{2Td}{T})$$

$$C = k_p \frac{Td}{T}$$

4

T A B C

4

k_p

k_i

k_i 0

k_p

15-a

k_i

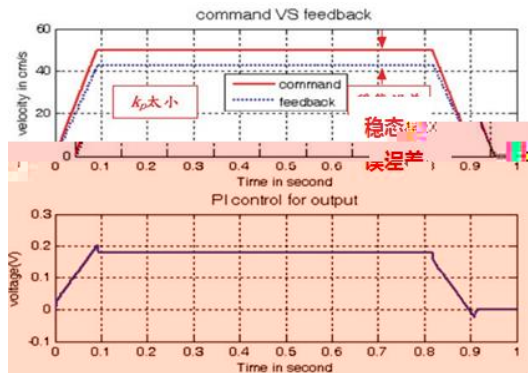
15-b

k_p

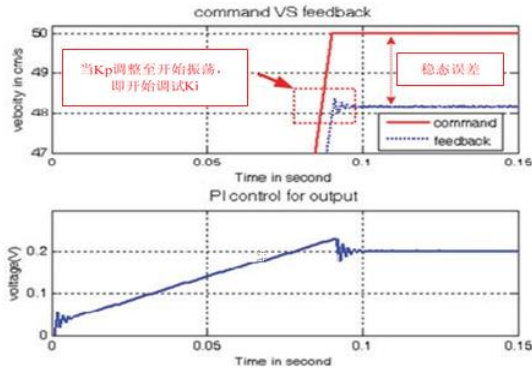
15-c

Feedback

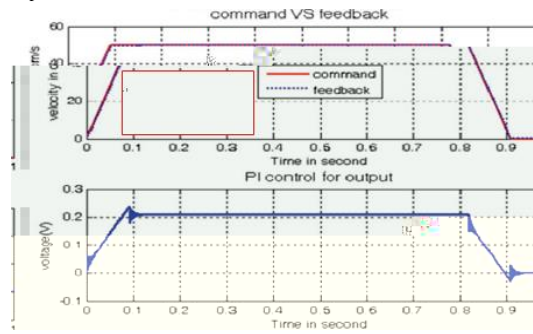
command



a. k_p 0.025 k_i 0 PI



b. k_p 0.011 k_i 0 PI



c. k_p 0.011 k_i 3 PI

15

2

16

PI

PI

xSpeed

$$2 \text{ rightSpeed} + \text{leftSpeed} / 2$$

$$\text{xError} \quad \text{xError} = \text{xSpeed} - \text{R} + \text{L} / 2$$

xError

PI

U1

PI

U1

PWM

xError

0

$$\text{xSpeed} = \text{R} + \text{L} / 2$$

PI

PI

wSpeed

$$2 \text{ rightSpeed} - \text{leftSpeed} / 2$$

$$\text{wError} \quad \text{wError} = \text{wSpeed} - \text{R} - \text{L} / 2$$

wError

PI

U2

PI

U2

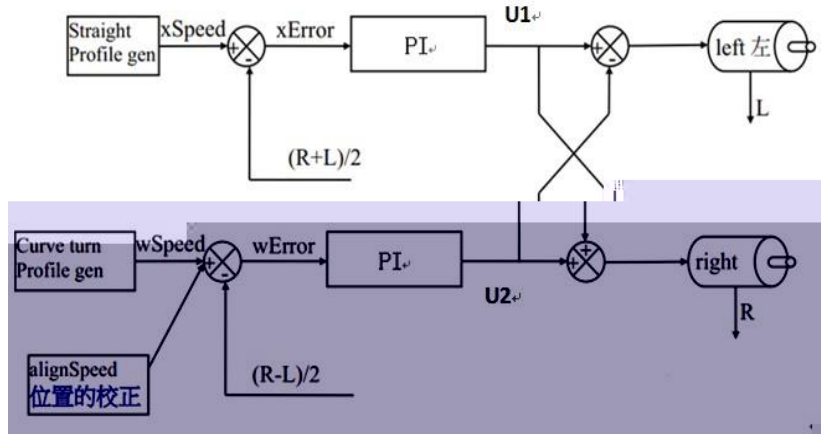
PWM

wError

0

wSpeed= (R-L) / 2

PI



16

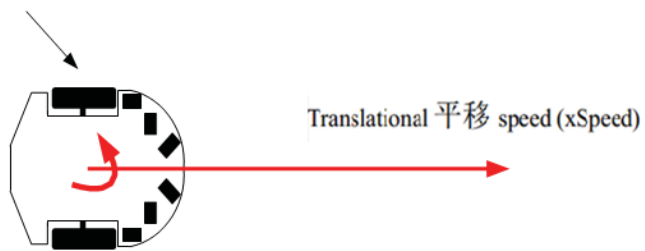
$$\text{leftSpeed} = \text{xSpeed} - \text{wSpeed}$$

$$\text{rightSpeed} = \text{xSpeed} + \text{wSpeed}$$

$$\text{xSpeed} = (\text{rightSpeed} + \text{leftSpeed}) / 2$$

$$\text{wSpeed} = (\text{rightSpeed} - \text{leftSpeed}) / 2$$

Rotational 旋转 speed (wSpeed)



17

U1

U2

U1

U2

17

PI

PI

PWM

e

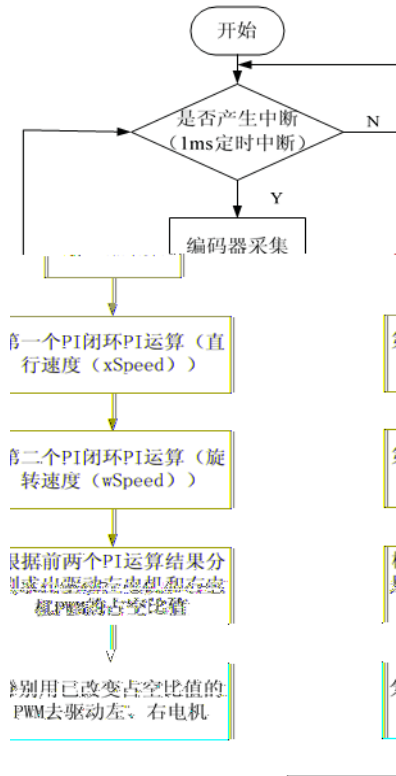
t1

t1

T

PWM

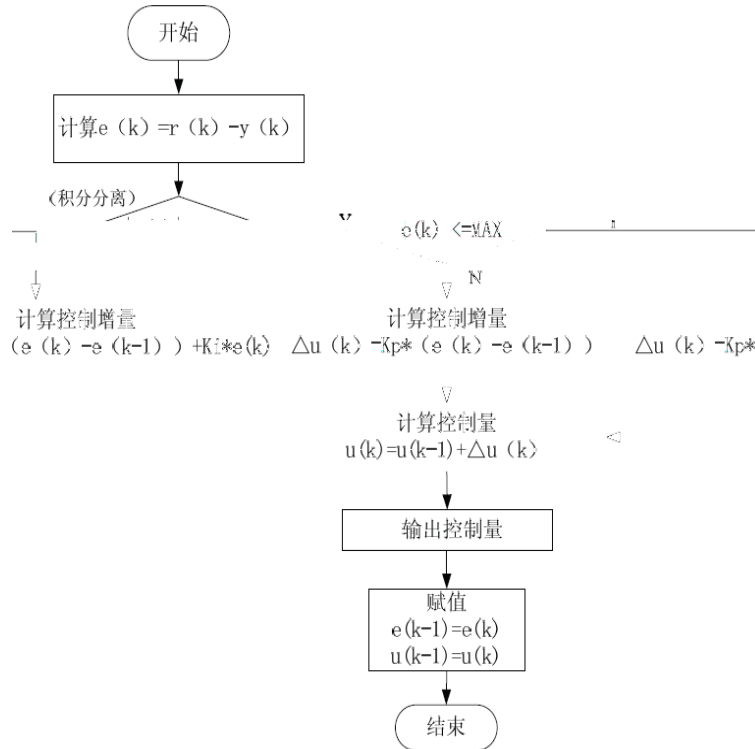
18



18

PID

19

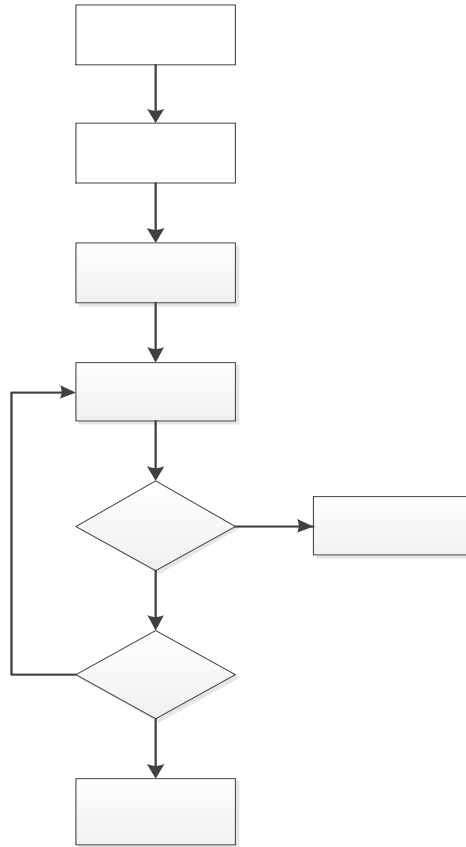


19

PID

4 2 2
1

20



20

2

```
void __sensorInit(void)
{
    TimerDisable(TIMER1_BASE,TIMER_B);
    TimerControlLevel(TIMER1_BASE,TIMER_B,true);
    TimerDisable(TIMER1_BASE,TIMER_B
        __IRSEND3_LR);
    PWMGenConfigure(PWM_BASE,PWM_GEN_1,PWM_GEN_MODE_UP_DOWN
        __IRSEND3_LR);
    PWMOinPutState(PWM_BASE,PWM_OUT_7_BIT
```



```

{
int16 Val;

TimerControlLevel (TIMER1_BASE,TIMER_B,true _MAIN
                    |__IRSEND3_LR);

TimerControlLevel(TIMER1_BASE,TIMER_B,true);

TimerDisable(TIMER1_BASE,TIMER_B);

zlg7289Download(0,0,0,GW%10);

mazeInit();

MPU6050_init(1);

zlg7289Download(0,0,0,GW%10)
{
    GYRO_Z_Angle();

    zlg7289Download(0,3,0,GW%10);

    zlg7289Download(0,0,0,GW%10);

    zlg7289Download(0,0,0,GW%10);

    zlg7289Download(0,3,0,GW%10);

    mazeInit();

}
}

4 2 4

    1

```

0

1

2

3

2

3

3

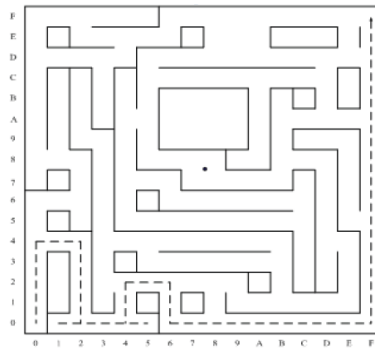
bit0	0	1	0
bit1	1	1	0
bit2	2	1	0
bit3	3	1	0

3

1

21

0 0

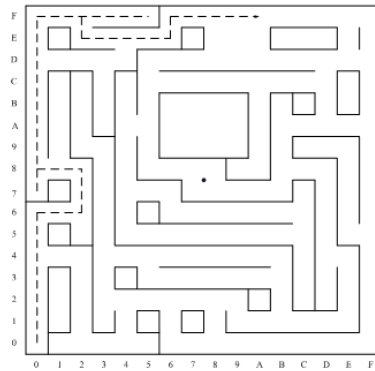


21

2

22

0 0

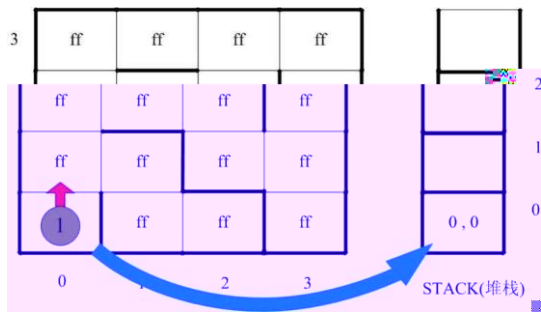


22

3

23

1 2 3

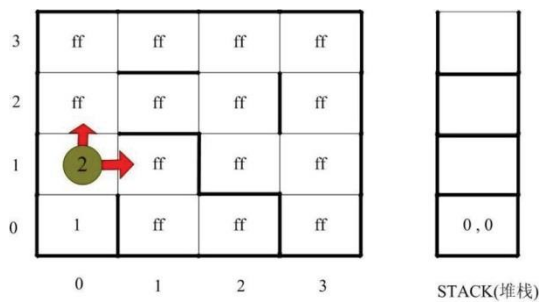


25

25

0 0

0 1 Step=Step+1 26



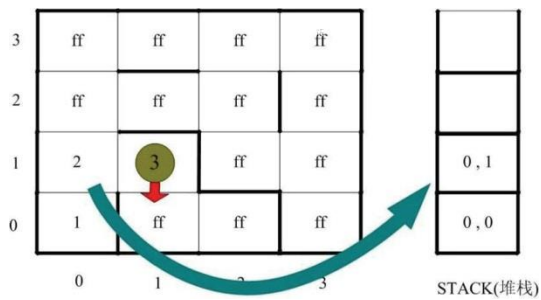
26

26

0 1

1 1 Step = Step + 1

27



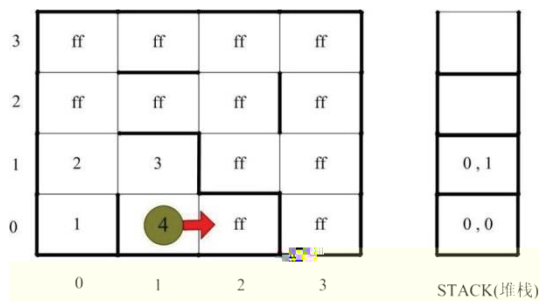
27

27

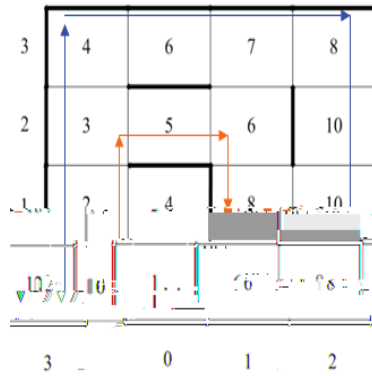
1 0

Step=Step+1

28



28



32

4 2 5

1

1

0

PI

$R+L / 2$

PI

PI

xError

0 xSpeed= $R+L / 2$

PI

wError

0

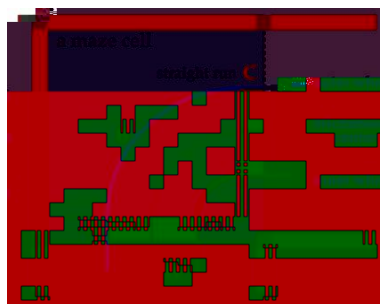
wSpeed= $R-L / 2=0$

R=L

xSpeed

2

33



33

90°

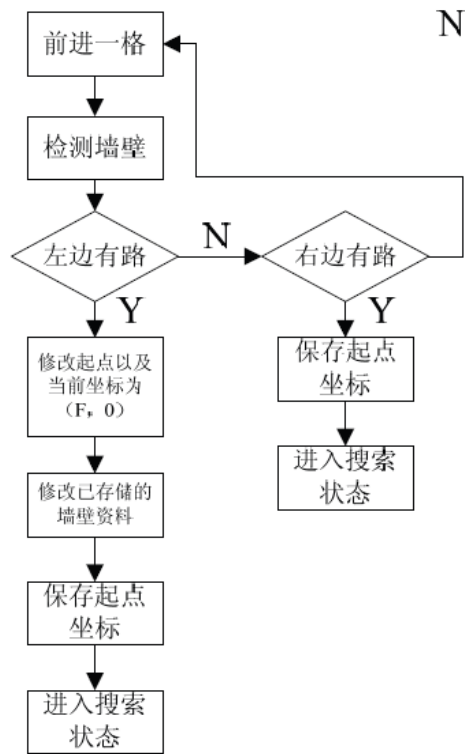
PI

		PI		
xSpeed	R+L /2	PI		
	R-L /2	PI	xError	0
xSpeed=	R+L /2=	PI	wError	0
wSpeed=	R-L /2=			R>L>0

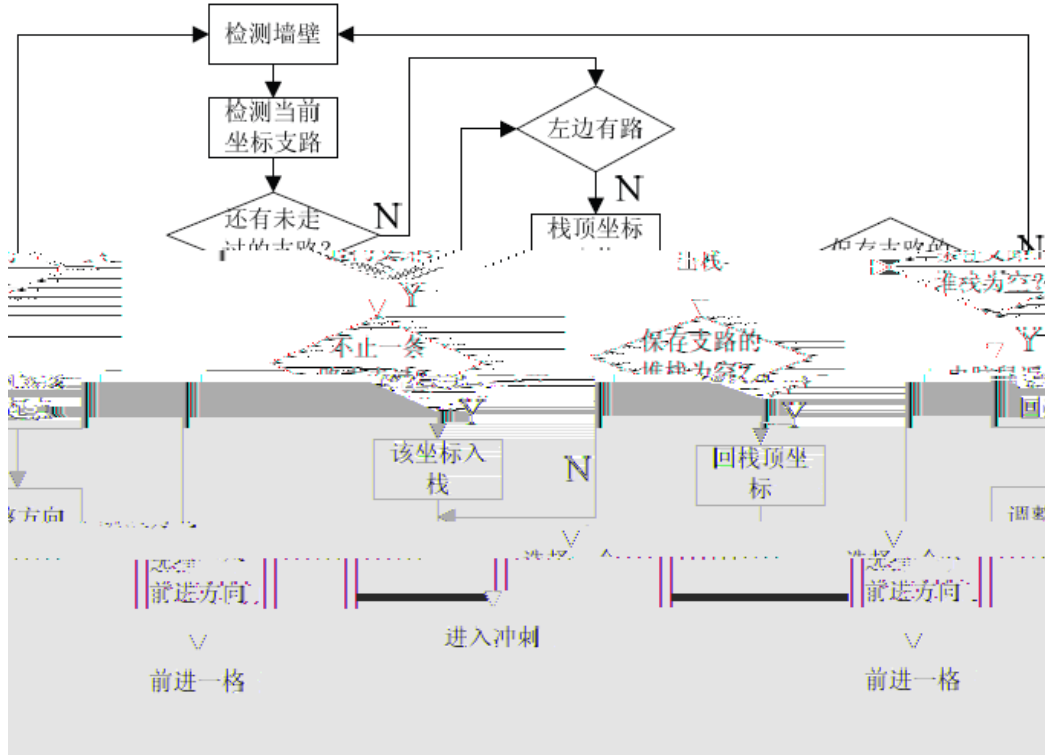
```

void mouseTurnleft(void)
{
    while((GmRight.uiPulseCtr+200)<=GmRight.uiPulse);
    while((GmLeft.uiPulseCtr+200 )<=GmLeft.uiPulse);
    GmLeft.uiPulse =3000;
    GmLeft.uiPulseCtr=0;
    GmRight.uiPulse =3000;
    GmRight.uiPulseCtr=0;
    mazeInit();
    break;
    GucMouseState=TURNLEFT;
    GucMouseDir=(GucMouseDir+3)%4;
    while(1)
    {
        if(GPIOPinRead(GPIO_PORTD_BASE, __DIR1) == 0)
        {
            break;
        }
    }
    while((GmRight.uiPulseCtr+200)<=GmRight.uiPulse);
    while((GmLeft.uiPulseCtr+200)<=GmLeft.uiPulse);
}

```

35

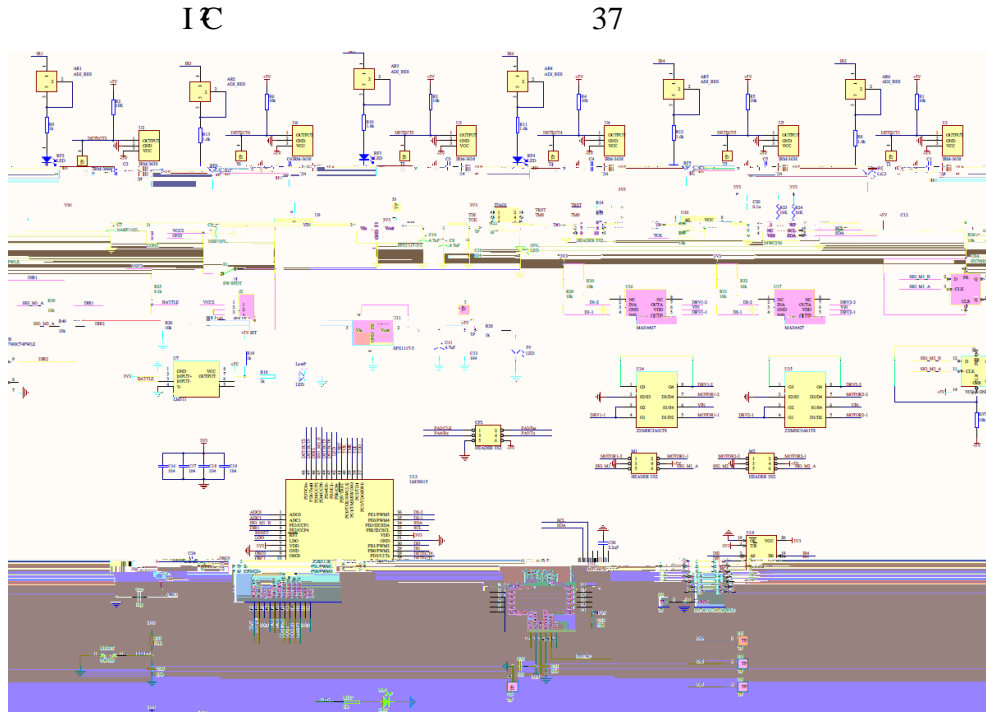


36

5

5 1

ARM LM3S615 PWM GPIO SPI /



37

PWM D1_1 D1_2 D2_1 D2_2 M4427 INB
 INA D1_1 D1_2 M4427
 INB INA D2_1 D2_2
 E D
 SIG_M1_A SIG_M2_A SIG_M1_B SIG_M2_B D CLK
 D
 74LS245 4 3 1 2 5 6 6
 45° 45° B1 C6 B0
 74LS245 6
 B4 B5 D7 D6 D3 D2 5 6 2 3 4 1
 SDA SCL FSYNC GPIOB3 GPIOB2 GPIOC4

MPU6050 SCL SDA FSYNC
 A1 A0 AT24C08 SDA SCL
 GPIOB3 GPIOB2 RESET C7 C5

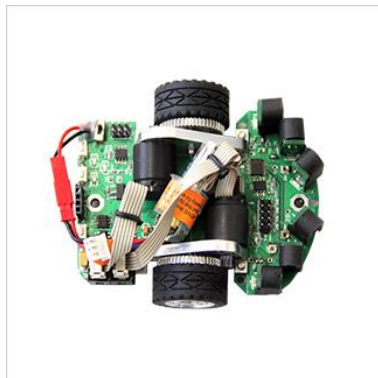
38 39 40 41



38 ARM



39 ARM



40 ARM

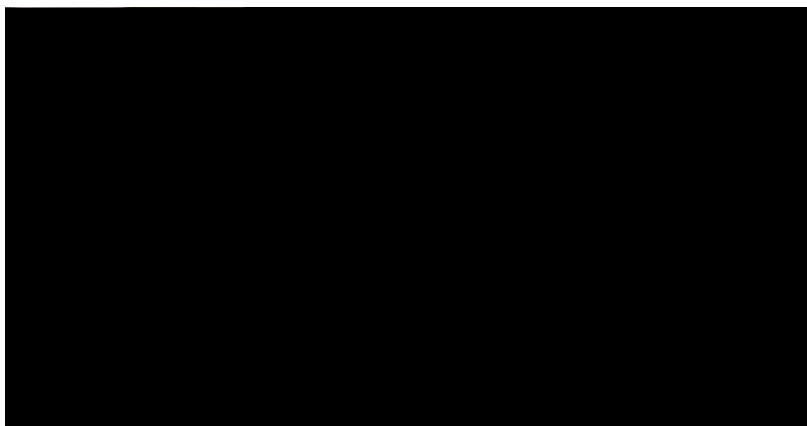


41 ARM

IAR EWARM

Build

42



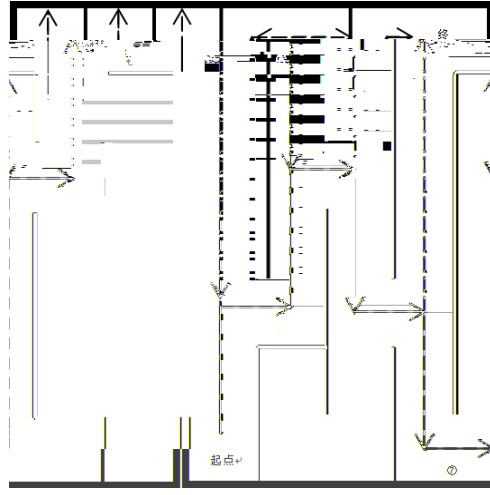
42 Build

5 2

ARM

43

4



43

4

-
-
- 1
 - 2
 - 3
 - 4
 - 5
 - 6
 - 7
-

5 3

1

ARM

ARM

7*7

3*3

1

3*4

2

4*3

3

4*4

4

1

2

3

4

ARM

Step

ARM

0 0

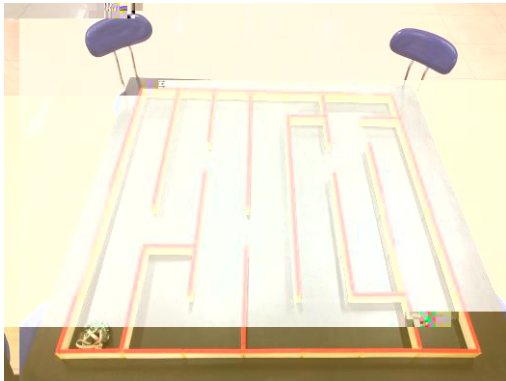
6 6

ARM

0 0

44

45



44

45

45

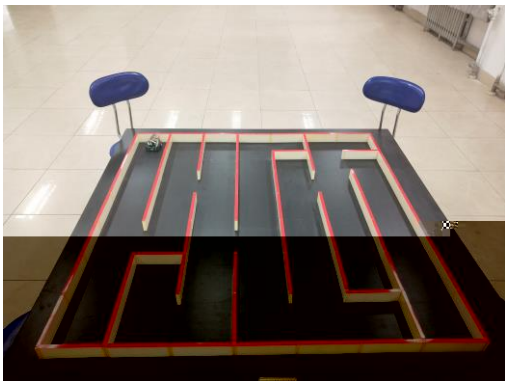
1

0 6

46

0 3

47



46

47

1 6

1 4

3

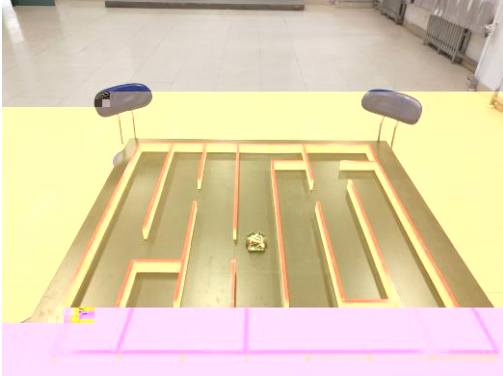
48



48

2 2
2

3 0



49

49 3 2

3 6

5 0

50



50

51

4 5



51

4

4

52



52

53

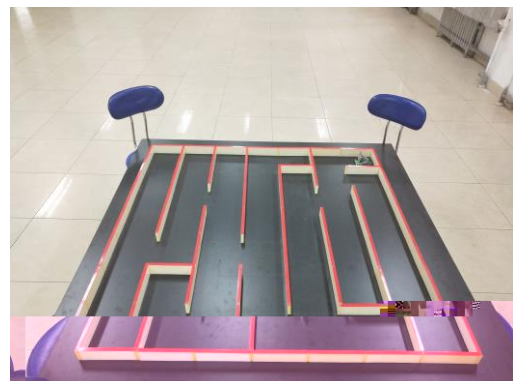
6 6



53

4

54



54

2

ARM

ARM

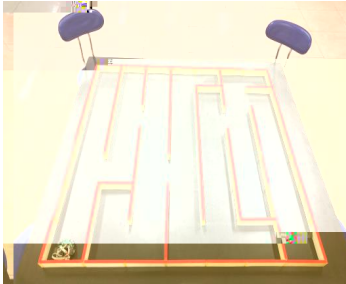
0 0

55

0 2

56

57



55

56

57

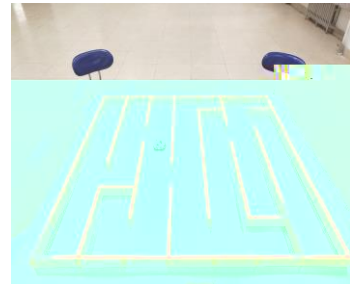
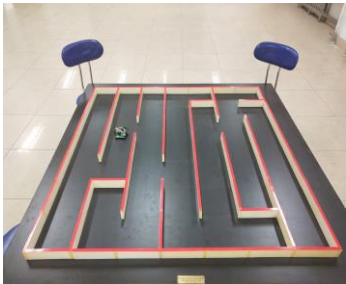
ARM

1 4

58

59

60



58

59

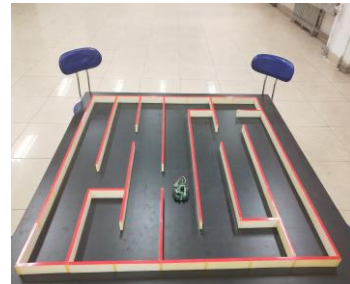
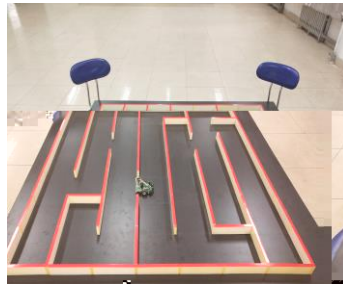
60

2 2

61

62

63



61

62

63

3 0

ARM

64

65

4 4

66



64



65



66

67

68

5 5

69

70



67



68



69



70

6

ARM

PID

C

1	Cortex-M3	LM3S615	
2		1717SR	1024
3			
4	MPU-6050		
5			

ARM

- [1] . [M]. : ,2016:1 3
- [2] , , . [J]. . 2017,24(06): 346
- [3] . ARM [D]. ,2014: 10 15
- [4] , . STC89C52 GPS [J]. . 2018(06): 36 40
- [5] , . MK60 [J]. (). 2017,12(04): 80 85
- [6] . LM3S615 [DB]. <https://www.docin.com/p-377162029.html>, 2007-03-31
- [7] , , . [J]. . 2018,41(18): 162 165+169
- [8] DR.FRITZ FAULHABER GMBH. EN_1717_SR_DFF[EB/OL].Technical Information, 2011- 2012
- [9] . ARM [J]. . 2018,45(03): 136 137
- [10] , . [J]. . 2012,34(22): 87 89+94
- [11] Everlight Electronics Co.,Ltd. IRM-8601S[EB/OL].<http://www.everlight.com>, 2004- 08- 19
- [12] , , . ARM Linux [J]. . 2018,27(08): 241 246
- [13] , , . ARM [J]. . 2010,39(01): 21 23+105
- [14] InvenSense Inc. MPU6050[EB/OL].<http://www.invensense.com>, 2011- 10- 24
- [15] , , . [J]. . 2010,18(02): 357 359+362
- [16] . [D]. ,2012: 2 35
- [17] . ARM [D]. ,2012: 35 45
- [18] Maxim Integrated Products. max4427[EB/OL].<http://www.alldatasheet.com>, 1993- 03- 01

[19] TEXAS INSTRUMENTS. cd74hct74[EB/OL].<http://www.ti.com>, 2003- 09

[20] Zetex Semiconductors plc. zxmhc3a01t8[EB/OL].<http://www.zetex.com>, 2004

[21] . [J]. . 2016(06): 73 74

[22] , , , . [J]. . 2016,33(02):
198 203+442

[23] TEXAS INSTRUMENTS. SN74LVC245A[EB/OL].<http://www.ti.com>, 2013- 03

[24] , . [J]. . 2017(05): 62 64

[25] Sipex Corporation. SPX1117[EB/OL].<http://www.maxlinear.com>, 2004- 11- 29

[26] , , . [J]. . 2018,25(10):
36 38+49

