

/\*

\* Atmega16codes.c

\* 1.LED glow

\* Created: 05-07-2019 05:52:38 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRB = 0xff;

while(1)

{

PORTB = 0x01;

}

}

/\*

\* Atmega16codes.c

\* 2.LED blynk

\* Created: 05-07-2019 05:52:38 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRB = 0xff;

while(1)

{

PORTB = 0x01;

\_delay\_ms(500);

PORTB = 0x00;

\_delay\_ms(500);

}



/\*

\* Atmega16codes.c

\* 3. 8-LED glow pattern 1

\* Created: 05-07-2019 05:52:38 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRB = 0xff;

while(1)

{

PORTB= 0x0f;

\_delay\_ms(500);

PORTB = 0xf0;

\_delay\_ms(500);

}

}

/\*

\* Atmega16codes.c

\* 4. 8-LED glow pattern 2

\* Created: 05-07-2019 05:52:38 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRB = 0xff;

while(1)

{

PORTB = 0x00;

int c =0;

for (int i = 1; i <= 128;i = i\*2)

{

PORTB = i + c;

\_delay\_ms(500);

c = PORTB;

}

}

}

/\*

\* Atmega16codes.c

\* 5. 8-LED glow pattern 3

\* Created: 05-07-2019 05:52:38 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRB = 0xff;

while(1)

{

PORTB = 0x00;

for ( int j = 16, i = 8; j<=128, i>= 1 ; j = j\*2, i = i/2)

{

PORTB = j+i;

\_delay\_ms(1000);

}

}

}

/\*

\* Atmega16codes.c

\* 6. 8-LED glow pattern 4

\* Created: 05-07-2019 05:52:38 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRB = 0xff;

while(1)

{

PORTB = 0x00;

int c = 0;

for ( int j = 128, i = 1; j>=16, i<= 8 ; j = j/2, i = i\*2)

{

PORTB = j+i+c;

\_delay\_ms(1000);

c = PORTB;

}

}

}

/\*

\* Atmega16codes.c

\* 7. 8-LED glow pattern 5

\* Created: 05-07-2019 05:52:38 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRB = 0xff;

while(1)

{

PORTB = 0x00;

for ( int j = 16, i = 8; j<=128, i>= 1 ; j = j\*2, i = i/2)

{

PORTB = j+i;

\_delay\_ms(1000);

}

}

}



/\*

\* Atmega16codes.c

\* 8. LED Matrix glow pattern T

\* Created: 05-07-2019 05:52:38 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRC = DDRD = 0xff;

while(1)

{

PORTC = 0b00011000;

PORTD = 0b00000000;

\_delay\_ms(40);

PORTD = 0b11111100;

PORTC = 0b11111111;

\_delay\_ms(40);

}

}



/\*

\* Atmega16codes.c

\* 9. LED Matrix glow pattern G

\* Created: 05-07-2019 05:52:38 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRC = DDRD = 0xff;

while(1)

{

PORTC = 0b00000011;

PORTD = 0b00000000;

\_delay\_ms(8);

PORTD = 0b00111111;

PORTC = 0b11111111;

\_delay\_ms(8);

PORTC = 0b11111111;

PORTD = 0b11111100;

\_delay\_ms(8);

PORTD = 0b11100111;

PORTC = 0b11111011;

\_delay\_ms(8);

PORTC = 0b11000000;

PORTD = 0b00000100;

\_delay\_ms(8);

}

}



/\*

\* Atmega16codes.c

\* 10. 7 segment display 1

\* Created: 05-07-2019 10:32:33 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRC= 0xff;

int n[10] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};

while(1)

{

for (int i = 0; i<=9 ; i++)

{

PORTC = n[i]; \_delay\_ms(1000);

}

}

}



/\*

\* Atmega16codes.c

\* 11. 7 segment display 2

\* Created: 05-07-2019 10:32:33 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRC = DDRD = 0xff;

while(1)

{

// for 14

PORTD = 0b00000010;

PORTC = 0b00000110;

\_delay\_ms(10);

PORTD = 0b00000001;

PORTC = 0b01100110;

\_delay\_ms(10);

}

}

/\*

\* Atmega16codes.c

\* 12. 7 segment display 3

\* Created: 05-07-2019 10:32:33 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRC = DDRD = 0xff;

int n[10] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};

while(1)

{

for (int j =0; j<=5;j++)

{

for (int i =0; i<=9;i++)

{

for (int p =1; p<=48;p++)

{

PORTD = 0b00000010;

PORTC = n[j];\_delay\_ms(10);

PORTD = 0b00000001;

PORTC = n[i];\_delay\_ms(10);

}

}

}

}

}



/\*

\* Atmega16codes.c

\* 13. 7 segment display 4

\* Created: 05-07-2019 10:32:33 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRC = DDRD = 0xff;

int a[10] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};

while(1)

{

for (int l =0; l<=9;l++)

{

for (int k = 0;k<=9;k++)

{

for (int j =0; j<=9;j++)

{

for (int i =0; i<=9;i++)

{

for (int p =1; p<=25;p++)

{

PORTD = 0b00001110;

PORTC = a[l]; \_delay\_ms(10);

PORTD = 0b00001101;

PORTC = a[k]; \_delay\_ms(10);

PORTD = 0b00001011;

PORTC = a[j]; \_delay\_ms(10);

PORTD = 0b00000111;

PORTC = a[i]; \_delay\_ms(10);

}

}

}

}

}

}

}



/\*

\* Atmega16codes.c

\* 14. Digital Watch 24 Hour time duration

\* accuracy 00:10:00 ± 00:00:02

\* Created: 05-07-2019 11:32:47 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRC = DDRD = 0xff;

int a[10] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};

while(1)

{

for (int n =0; n<=2;n++)

{

for (int m = 0; m<=3;m++)

{

for (int l =0; l<=5;l++)

{

for (int k = 0;k<=9;k++)

{

for (int j =0; j<=5;j++)

{

for (int i =0; i<=9;i++)

{

for (int p =1; p<=16;p++)

{

PORTD = 0b00111110;

PORTC = a[n];

\_delay\_ms(7);

PORTD = 0b00111101;

PORTC = a[m] + 0x80;

\_delay\_ms(7);

PORTD = 0b00111011;

PORTC = a[l];

\_delay\_ms(9);

PORTD = 0b00110111;

PORTC = a[k] + 0x80;

\_delay\_ms(9);

PORTD = 0b00101111;

PORTC = a[j];

\_delay\_ms(9);

PORTD = 0b00011111;

PORTC = a[i];

\_delay\_ms(9);

}

}

}

}

}

}

}

}

}



/\*

\* Atmega16codes.c

\* 15. Traffic Light Control system 29 sec.

\* Created: 06-07-2019 10:29:17 AM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

int main(void)

{

DDRC = DDRD = DDRB = 0xff;

int a[10] = {0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};

int b = 0b00100001, c = 0b00001100;

while (1)

{

for (int j = 0; j<=2; j++)

{

for (int i =0; i<=9; i++)

{

for (int p = 0; p<=50; p++) //50\*(10+10)ms = 1000ms

{

PORTD = 0b00000010;

PORTC = a[j]; \_delay\_ms(10);

PORTD = 0b00000001;

PORTC = a[i]; \_delay\_ms(10);

}

if (i==9 & j==2)

{

PORTB = b;

b = c;

c = PORTB;

}

}

}

}

}

8bit LCD mode



## Simulation design Atmega16 with LM016L LCD display 8bit

1. Code for display a normal character

/\*

\* LCD16x2.c

\*

\* Created: 10-06-2019 03:35:01 PM

\* Author: gulshan

\* Display a normal data

\*/

#include <avr/io.h>

#include <util/delay.h>

lcd\_init()

{

lcd\_cmd(0x02);

lcd\_cmd(0x38);

lcd\_cmd(0x0c);

}

lcd\_cmd(int x)

{

PORTD = x;

PORTC = 0b00000100;

\_delay\_ms(2);

PORTC = 0b00000000;

}

lcd\_data(int y)

{

PORTD = y;

PORTC = 0b00000101;

\_delay\_ms(2);

PORTC = 0b00000001;

}

int main(void)

{

while(1)

{

DDRC = DDRD = 0xff;

lcd\_init();

lcd\_cmd(0x81);

lcd\_data('G');

\_delay\_ms(8);

}

}

1. Code for a moving character

/\*

\* Gmove.c

\*

\* Created: 10-06-2019 04:12:07 PM

\* Author: gulshan

\*/

#include <avr/io.h>

#include <util/delay.h>

lcd\_init()

{

lcd\_cmd(0x02);

lcd\_cmd(0x38);

lcd\_cmd(0x0c);

}

lcd\_cmd(int x)

{

PORTD = x;

PORTC = 0b00000100;

\_delay\_ms(2);

PORTC = 0b00000000;

}

lcd\_data(int y)

{

PORTD = y;

PORTC = 0b00000101;

\_delay\_ms(2);

PORTC = 0b00000001;

}

int main(void)

{

while(1)

{

DDRC = DDRD = 0xff;

for ( int i = 0x80; i <=0x8f; i++)

{

lcd\_init();

lcd\_cmd(i);

lcd\_data('G');

\_delay\_ms(200);

lcd\_cmd(i);

lcd\_data(' ');

\_delay\_ms(20);

}

}

}

1. Code for String and Number for 8bit

/\*

\* string\_and\_num.c

\*

\* Created: 11-06-2019 03:44:36 AM

\* Author: gulshan

\*/

#include <avr/io.h>

#include <util/delay.h>

lcd\_init()

{

lcd\_cmd(0x02);

lcd\_cmd(0x38);

lcd\_cmd(0x0c);

}

// Function for cmd

lcd\_cmd(int x)

{

PORTD = x;

PORTC = 0b00000100;

\_delay\_ms(2);

PORTC = 0b00000000;

}

// Function for data

lcd\_data(int y)

{

PORTD = y;

PORTC = 0b00000101;

\_delay\_ms(2);

PORTC = 0b00000001;

}

// function for string

lcd\_string(char \*str)

{

int i = 0;

while(str[i] != '\0')

{

lcd\_data(str[i]);

i++;

lcd\_cmd(0x06);

}

}

// function for numbers

lcd\_num( int num)

{

int d = 0;

lcd\_cmd(0x04);

while( num != 0)

{

d = num % 10;

lcd\_data(d+48);

num = num/10;

}

}

int main(void)

{

DDRC = DDRD = 0xff;

lcd\_init();

while(1)

{

lcd\_cmd(0x01);

\_delay\_ms(10);

lcd\_cmd(0x82);

lcd\_string("Hello Warld");

\_delay\_ms(1000);

lcd\_cmd(0xc1);

lcd\_num(34);

\_delay\_ms(1000);

}

}

1. Code for display name using array

/\*

\* numbermove.c

\*

\* Created: 11-06-2019 02:30:55 PM

\* Author: gulshan

\*/

#include <avr/io.h>

#include <util/delay.h>

lcd\_init()

{

lcd\_cmd(0x02);

lcd\_cmd(0x38);

lcd\_cmd(0x0c);

}

lcd\_cmd(int x)

{

PORTD = x;

PORTC = 0b00000100;

\_delay\_ms(2);

PORTC = 0b00000000;

}

lcd\_data(int y)

{

PORTD = y;

PORTC = 0b00000101;

\_delay\_ms(2);

PORTC = 0b00000001;

}

int main(void)

{

int a[5] = {'H','E','L','L','O'};

while(1)

{

DDRC = DDRD = 0xff;

for ( int i = 0x80, j = 0; i <=0x8f, j <=4 ; i++, j++)

{

lcd\_init();

lcd\_cmd(i);

lcd\_data(a[j]);

\_delay\_ms(500);

}

}

}

1. LCD clock

/\*

\* LCDclock.c

\*

\* Created: 11-06-2019 03:44:36 AM

\* Author: gulshan

\*/

#include <avr/io.h>

#include <util/delay.h>

lcd\_init()

{

lcd\_cmd(0x02);

lcd\_cmd(0x38);

lcd\_cmd(0x0c);

}

// Function for cmd

lcd\_cmd(int x)

{

PORTD = x;

PORTC = 0b00000100;

\_delay\_ms(2);

PORTC = 0b00000000;

}

// Function for data

lcd\_data(int y)

{

PORTD = y;

PORTC = 0b00000101;

\_delay\_ms(2);

PORTC = 0b00000001;

}

// function for string

lcd\_string(char \*str)

{

int i = 0;

while(str[i] != '\0')

{

lcd\_data(str[i]);

i++;

lcd\_cmd(0x06);

}

}

// function for numbers

lcd\_num( int num)

{

int d = 0;

lcd\_cmd(0x04);

while( num != 0)

{

d = num % 10;

lcd\_data(d+48);

num = num/10;

}

}

int main(void)

{

while(1)

{

DDRC = DDRD = 0xff;

lcd\_cmd(0x81);

lcd\_string("00:00:00");

for( int g = 1 ; g <= 24 ; g++)

{

for (int m = 1; m <= 59; m++)

{

for (int s=0; s <= 59; s++)

{

lcd\_init();

lcd\_cmd(0x88);

lcd\_num(s);

\_delay\_ms(782);

}

lcd\_cmd(0x88);

lcd\_string("00");

lcd\_cmd(0x85);

lcd\_num(m);

}

lcd\_cmd(0x85);

lcd\_string("00");

lcd\_cmd(0x82);

lcd\_num(g);

}

}

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4bit LCD mode



## Simulation design Atmega16 with LM016L LCD display 4bit

1. Code for display a normal character in 4bit mode

/\*

\* \_4bitlcd.c

\*

\* Created: 12-06-2019 02:57:34 PM

\* Author: gulshan

\*/

#include <avr/io.h>

#include <util/delay.h>

#define lcd PORTC

lcd\_4bitcmd(int x)

{

lcd = x & 0xf0;

lcd = lcd+0x04;

\_delay\_ms(2);

lcd = lcd-0x04;

lcd = (x<<4) & 0xf0;

lcd = lcd+0x04;

\_delay\_ms(2);

lcd = lcd - 0x04;

}

lcd\_4bitdata(int y)

{

lcd = y & 0xf0;

lcd = lcd+0x05;

\_delay\_ms(2);

lcd = lcd-0x04;

lcd = (y<<4) & 0xf0;

lcd = lcd + 0x05;

\_delay\_ms(2);

lcd = lcd - 0x04;

}

lcd\_4bitinit()

{

lcd\_4bitcmd(0x02);

lcd\_4bitcmd(0x28);

lcd\_4bitcmd(0x0c);

}

int main(void)

{

while(1)

{

DDRC = 0xff;

lcd\_4bitinit();

lcd\_4bitcmd(0x85);

lcd\_4bitdata('i');

}

}

1. Code for string and Number in 4bit mode

/\*

\* \_4bitlcd.c

\*

\* Created: 12-06-2019 02:57:34 PM

\* Author: gulshan

\*/

#include <avr/io.h>

#include <util/delay.h>

#define lcd PORTC

lcd\_4bitcmd(int x)

{

lcd = x & 0xf0;

lcd = lcd+0x04;

\_delay\_ms(2);

lcd = lcd-0x04;

lcd = (x<<4) & 0xf0;

lcd = lcd+0x04;

\_delay\_ms(2);

lcd = lcd - 0x04;

}

lcd\_4bitdata(int y)

{

lcd = y & 0xf0;

lcd = lcd+0x05;

\_delay\_ms(2);

lcd = lcd-0x04;

lcd = (y<<4) & 0xf0;

lcd = lcd + 0x05;

\_delay\_ms(2);

lcd = lcd - 0x04;

}

lcd\_4bitinit()

{

lcd\_4bitcmd(0x02);

lcd\_4bitcmd(0x28);

lcd\_4bitcmd(0x0c);

}

lcd\_4bitstring(char \*str)

{

int i = 0;

while(str[i] != '\0')

{

lcd\_4bitdata(str[i]);

i++;

}

}

lcd\_4bitnum( int num)

{

int d = 0;

lcd\_4bitcmd(0x04);

while( num != 0)

{

d = num % 10;

lcd\_4bitdata(d+48);

num = num/10;

}

lcd\_4bitcmd(0x06);

}

int main(void)

{

DDRC = 0xff;

lcd\_4bitinit();

while(1)

{

lcd\_4bitcmd(0x85); // string

lcd\_4bitstring("LOVE PUBG");

lcd\_4bitcmd(0xc8); // number

lcd\_4bitnum(3000);

\_delay\_ms(1000);

}

}

1. Character in 4bit mode

/\*

\* character.c

\*

\* Created: 12-06-2019 04:12:11 PM

\* Author: gulshan

\*/

#include <avr/io.h>

#include <util/delay.h>

#define lcd PORTC

lcd\_4bitcmd(int x)

{

lcd = x & 0xf0;

lcd = lcd+0x04;

\_delay\_ms(1);

lcd = lcd-0x04;

lcd = (x<<4) & 0xf0;

lcd = lcd+0x04;

\_delay\_ms(1);

lcd = lcd - 0x04;

}

lcd\_4bitdata(int y)

{

lcd = y & 0xf0;

lcd = lcd+0x05;

\_delay\_ms(1);

lcd = lcd-0x04;

lcd = (y<<4) & 0xf0;

lcd = lcd + 0x05;

\_delay\_ms(1);

lcd = lcd - 0x04;

}

lcd\_4bitinit()

{

lcd\_4bitcmd(0x02);

lcd\_4bitcmd(0x28);

lcd\_4bitcmd(0x0c);

}

void spcl1()

{

lcd\_4bitcmd(0x40);

lcd\_4bitdata(0b11111111);

lcd\_4bitdata(0b11111111);

lcd\_4bitdata(0b11111111);

lcd\_4bitdata(0b11111111);

lcd\_4bitdata(0b11111111);

lcd\_4bitdata(0b11111111);

lcd\_4bitdata(0b11111111);

}

void spcl2()

{

lcd\_4bitcmd(0x48);

lcd\_4bitdata(0b00001010);

lcd\_4bitdata(0b00010101);

lcd\_4bitdata(0b00010001);

lcd\_4bitdata(0b00001010);

lcd\_4bitdata(0b00000100);

lcd\_4bitdata(0b00000000);

lcd\_4bitdata(0b00000000);

}

int main(void)

{

while(1)

{

DDRC = 0xff;

lcd\_4bitinit();

spcl2();

lcd\_4bitcmd(0xc5);

lcd\_4bitdata(0);

lcd\_4bitinit();

spcl1();

lcd\_4bitcmd(0x85);

lcd\_4bitdata(1);

}

}

1. Animation in 4bit mode (homework)

/\*

\* animation.c

\*

\* Created: 13-06-2019 10:29:33 AM

\* Author: gulshan

\*/

#include <avr/io.h>

#include <util/delay.h>

#define lcd PORTC

lcd\_4bitcmd(int x)

{

lcd = x & 0xf0;

lcd = lcd+0x04;

\_delay\_ms(1);

lcd = lcd-0x04;

lcd = (x<<4) & 0xf0;

lcd = lcd+0x04;

\_delay\_ms(1);

lcd = lcd - 0x04;

}

lcd\_4bitdata(int y)

{

lcd = y & 0xf0;

lcd = lcd+0x05;

\_delay\_ms(1);

lcd = lcd-0x04;

lcd = (y<<4) & 0xf0;

lcd = lcd + 0x05;

\_delay\_ms(1);

lcd = lcd - 0x04;

}

// function for string

lcd\_4bitstring(char \*str)

{

int i = 0;

while(str[i] != '\0')

{

lcd\_4bitdata(str[i]);

i++;

}

}

lcd\_4bitinit()

{

lcd\_4bitcmd(0x02);

lcd\_4bitcmd(0x28);

lcd\_4bitcmd(0x0c);

}

void spcl1()

{

lcd\_4bitcmd(0x40);

lcd\_4bitdata(0b00000000);

lcd\_4bitdata(0b00001100);

lcd\_4bitdata(0b00011111);

lcd\_4bitdata(0b00001100);

lcd\_4bitdata(0b00000000);

lcd\_4bitdata(0b00000000);

lcd\_4bitdata(0b00000000);

}

void spcl2()

{

lcd\_4bitcmd(0x48);

lcd\_4bitdata(0b00011011);

lcd\_4bitdata(0b00011111);

lcd\_4bitdata(0b00011111);

lcd\_4bitdata(0b00001110);

lcd\_4bitdata(0b00000100);

lcd\_4bitdata(0b00000000);

lcd\_4bitdata(0b00000000);

}

int main(void)

{

DDRC = 0xff;

lcd\_4bitinit();

lcd\_4bitcmd(0x83);

lcd\_4bitstring("Welcome to");

lcd\_4bitcmd(0xc5);

lcd\_4bitstring("Play Wizard");

\_delay\_ms(1000);

while(1)

{

lcd\_4bitcmd(0x01);

\_delay\_ms(10);

for (int i = 0x8f; i>= 0x81; i--)

{

lcd\_4bitinit();

spcl2();

lcd\_4bitcmd(i);

lcd\_4bitdata(1);

\_delay\_ms(500);

lcd\_4bitcmd(i);

lcd\_4bitdata(' ');

\_delay\_ms(50);

}

}

}



/\*

\* Atmega16codes.c

\* Keypad Interfacing

\* Created: 07-07-2019 2:23:37 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

#define lcd PORTD

lcd\_4bitinit() {

lcd\_4bitcmd(0x02);

lcd\_4bitcmd(0x28);

lcd\_4bitcmd(0x0c);

}

lcd\_4bitstr(char \*str) {

int d=0;

while(str[d]!='\0')

{

lcd\_4bitdata(str[d]);

lcd\_4bitcmd(0x06);

d++;

} }

lcd\_4bitdata(int x) {

lcd=(x&0xf0)+0x05;

\_delay\_ms(2);

lcd=lcd-0x04;

lcd=((x<<4)&0xf0)+0x05;

\_delay\_ms(2);

lcd=lcd-0x04;

}

lcd\_4bitcmd(int x) {

lcd=(x&0xf0)+0x04;

\_delay\_ms(2);

lcd=lcd-0x04;

lcd=((x<<4)&0xf0)+0x04;

\_delay\_ms(2);

lcd=lcd-0x04;

}

lcd\_4bitnum(int n) {

int d;

if(n==0){lcd\_4bitdata(48);}

lcd\_4bitcmd(0x04);

while(n!=0)

{

d=n%10;

lcd\_4bitdata(d+48);

n=n/10;

} }

int key()

{

while (1) {

PORTB = 0b11011111; \_delay\_ms(20);

if (PINA == 0xfe) { while (PINA == 0xfe); return 1;}

if (PINA == 0xfd) { while (PINA == 0xfd); return 4;}

if (PINA == 0xfb) { while (PINA == 0xfb); return 7;}

if (PINA == 0xf7) { while (PINA == 0xf7); return 10;}

PORTB = 0b10111111; \_delay\_ms(20);

if (PINA == 0xfe) { while (PINA == 0xfe); return 2;}

if (PINA == 0xfd) { while (PINA == 0xfd); return 5;}

if (PINA == 0xfb) { while (PINA == 0xfb); return 8;}

if (PINA == 0xf7) { while (PINA == 0xf7); return 11;}

PORTB = 0b01111111; \_delay\_ms(30);

if (PINA == 0xfe) { while (PINA == 0xfe); return 3;}

if (PINA == 0xfd) { while (PINA == 0xfd); return 6;}

if (PINA == 0xfb) { while (PINA == 0xfb); return 9;}

if (PINA == 0xf7) { while (PINA == 0xf7); return 12;}

} }

int main(void)

{

DDRD = DDRB = 0xff;

DDRA = 0xf0;

PORTA = 0xff;

lcd\_init();

while(1)

{

int a = key();

if (a<10)

{ lcd\_4bitcmd(0x01); \_delay\_ms(10); }

lcd\_4bitcmd(0x83);

lcd\_number(a);

}

}

1. Passward in 4bit mode (homework)



/\*

\* Atmega16codes.c

\* Password code 3456

\* Created: 06-07-2019 11:39:07 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

#define lcd PORTD

lcd\_4bitcmd(int x)

{

lcd = x & 0xf0;

lcd = lcd+0x04;

\_delay\_ms(2);

lcd = lcd-0x04;

lcd = (x<<4) & 0xf0;

lcd = lcd+0x04;

\_delay\_ms(2);

lcd = lcd - 0x04;

}

lcd\_4bitdata(int y)

{

lcd = y & 0xf0;

lcd = lcd+0x05;

\_delay\_ms(2);

lcd = lcd-0x04;

lcd = (y<<4) & 0xf0;

lcd = lcd + 0x05;

\_delay\_ms(2);

lcd = lcd - 0x04;

}

lcd\_4bitinit()

{

lcd\_4bitcmd(0x02);

lcd\_4bitcmd(0x28);

lcd\_4bitcmd(0x0c);

}

lcd\_4bitnum(int num)

{

lcd\_4bitcmd(0x04);

int d = 0;

while (num != 0)

{

d = num%10;

lcd\_4bitdata(d+48);

num = num/10;

}

lcd\_4bitcmd(0x06);

}

int key()

{

while (1)

{

PORTB = 0b01111111;

\_delay\_ms(30);

if (PINA == 0xfe){ while (PINA == 0xfe); return 1;}

if (PINA == 0xfd)

{ while (PINA == 0xfd)

{

}

return 4;

}

if (PINA == 0xfb)

{while (PINA == 0xfb)

{

}

return 7;

}

if (PINA == 0xf7)

{while (PINA == 0xf7)

{

}

return 10;

}

PORTB = 0b10111111;

\_delay\_ms(30);

if (PINA == 0xfe)

{while (PINA == 0xfe)

{

}

return 2;

}

if (PINA == 0xfd)

{while (PINA == 0xfd)

{

}

return 5;

}

if (PINA == 0xfb)

{while (PINA == 0xfeb)

{

}

return 8;

}

if (PINA == 0xf7)

{while (PINA == 0xf7)

{

}

return 11;

}

PORTB = 0b11011111;

\_delay\_ms(30);

if (PINA == 0xfe)

{while (PINA == 0xfe)

{

}

return 3;

}

if (PINA == 0xfd)

{while (PINA == 0xfd)

{

}

return 6;

}

if (PINA == 0xfb)

{while (PINA == 0xfb)

{

}

return 9;

}

if (PINA == 0xf7)

{while (PINA == 0xf7)

{

}

return 12;

}

}

}

int main(void)

{

DDRC = DDRD = DDRB = 0xff;

DDRA = 0xf0;

PORTA = 0xff;

int pass = 3456;

int b = 0;

lcd\_4bitinit();

while(1)

{ for (int i = 0; i<4 ;i++)

{

int a = key();

b = b\*10 + a;

lcd\_4bitcmd(0x83);

lcd\_4bitnum(b);

}

if (b==pass)

{ PORTC = 0xff;

}

else

{ PORTC = 0x00;

} }

}



/\*

\* Atmega16codes.c

\* M0t0r control with Keypad

\* Created: 07-07-2019 11:04:07 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

#define lcd PORTD

lcd\_4bitinit() {

lcd\_4bitcmd(0x02);

lcd\_4bitcmd(0x28);

lcd\_4bitcmd(0x0c);

}

lcd\_4bitstr(char \*str) {

int d=0;

while(str[d]!='\0')

{

lcd\_4bitdata(str[d]);

lcd\_4bitcmd(0x06);

d++;

} }

lcd\_4bitdata(int x) {

lcd=(x&0xf0)+0x05;

\_delay\_ms(2);

lcd=lcd-0x04;

lcd=((x<<4)&0xf0)+0x05;

\_delay\_ms(2);

lcd=lcd-0x04;

}

lcd\_4bitcmd(int x) {

lcd=(x&0xf0)+0x04;

\_delay\_ms(2);

lcd=lcd-0x04;

lcd=((x<<4)&0xf0)+0x04;

\_delay\_ms(2);

lcd=lcd-0x04;

}

lcd\_4bitnum(int n) {

int d;

if(n==0){lcd\_4bitdata(48);}

lcd\_4bitcmd(0x04);

while(n!=0)

{

d=n%10;

lcd\_4bitdata(d+48);

n=n/10;

} }

int key()

{

while (1) {

PORTB = 0b11011111; \_delay\_ms(20);

if (PINA == 0xfe) { while (PINA == 0xfe); return 1;}

if (PINA == 0xfd) { while (PINA == 0xfd); return 4;}

if (PINA == 0xfb) { while (PINA == 0xfb); return 7;}

if (PINA == 0xf7) { while (PINA == 0xf7); return 10;}

PORTB = 0b10111111; \_delay\_ms(20);

if (PINA == 0xfe) { while (PINA == 0xfe); return 2;}

if (PINA == 0xfd) { while (PINA == 0xfd); return 5;}

if (PINA == 0xfb) { while (PINA == 0xfb); return 8;}

if (PINA == 0xf7) { while (PINA == 0xf7); return 11;}

PORTB = 0b01111111; \_delay\_ms(30);

if (PINA == 0xfe) { while (PINA == 0xfe); return 3;}

if (PINA == 0xfd) { while (PINA == 0xfd); return 6;}

if (PINA == 0xfb) { while (PINA == 0xfb); return 9;}

if (PINA == 0xf7) { while (PINA == 0xf7); return 12;}

} }

int main(void)

{

DDRC = DDRB = DDRD = 0xff;

DDRA = 0xf0;

PORTA = 0xff;

int n[5] = {2,8,4,6,0};

lcd\_4bitinit();

while(1)

{ int a = key();

lcd\_4bitcmd(0x01); \_delay\_ms(10);

lcd\_4bitcmd(0x83); \_delay\_ms(100);

lcd\_4bitnum(a);

if (a == n[0])

{ PORTC = 0b00001001;

lcd\_4bitcmd(0xc3);

lcd\_4bitstr(" Forward"); }

else if(a == n[1])

{ PORTC = 0b00000110;

lcd\_4bitcmd(0xc3);

lcd\_4bitstr(" Backward"); }

else if(a == n[2])

{ PORTC = 0b00000001;

lcd\_4bitcmd(0xc3);

lcd\_4bitstr(" Left"); }

else if(a == n[3])

{ PORTC = 0b00001000;

lcd\_4bitcmd(0xc3);

lcd\_4bitstr(" Right"); }

else

{ PORTC = 0b00000000;

lcd\_4bitcmd(0xc3);

lcd\_4bitstr(" Stop"); }

} }











/\*

\* Atmega16codes.c

\* Object Avoiding

\* Created: 08-07-2019 04:04:56 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

#define lcd PORTD

lcd\_4bitinit() {

lcd\_4bitcmd(0x02);

lcd\_4bitcmd(0x28);

lcd\_4bitcmd(0x0c);

}

lcd\_4bitstr(char \*str) {

int d=0;

while(str[d]!='\0')

{

lcd\_4bitdata(str[d]);

lcd\_4bitcmd(0x06);

d++;

}

}

lcd\_4bitdata(int x) {

lcd=(x&0xf0)+0x05;

\_delay\_ms(2);

lcd=lcd-0x04;

lcd=((x<<4)&0xf0)+0x05;

\_delay\_ms(2);

lcd=lcd-0x04;

}

lcd\_4bitcmd(int x) {

lcd=(x&0xf0)+0x04;

\_delay\_ms(2);

lcd=lcd-0x04;

lcd=((x<<4)&0xf0)+0x04;

\_delay\_ms(2);

lcd=lcd-0x04;

}

lcd\_4bitnum(int n) {

int d;

if(n==0){lcd\_4bitdata(48);}

lcd\_4bitcmd(0x04);

while(n!=0)

{

d=n%10;

lcd\_4bitdata(d+48);

n=n/10;

}

}

int main(void)

{

DDRC = DDRD = 0xff;

DDRA = 0xfe;

lcd\_4bitinit();

while(1)

{

lcd\_4bitcmd(0x83);

lcd\_4bitstr(" Obj. Avoid");

while(PINA == 0x01)

{

PORTC = 0x00;

lcd\_4bitcmd(0x01); \_delay\_ms(10);

lcd\_4bitcmd(0xc3);

lcd\_4bitstr(" Obj. Stop");

\_delay\_ms(500); //stop

PORTC = 0x06;

lcd\_4bitcmd(0x01); \_delay\_ms(10);

lcd\_4bitcmd(0xc3);

lcd\_4bitstr(" Backward");

\_delay\_ms(1000); //Backward

PORTC = 0x08;

lcd\_4bitcmd(0x01); \_delay\_ms(10);

lcd\_4bitcmd(0xc3);

lcd\_4bitstr(" Right turn");

\_delay\_ms(1000); //Right turn

lcd\_4bitcmd(0x01); \_delay\_ms(10);

}

PORTC = 0x09; //forward

lcd\_4bitcmd(0xc3);

lcd\_4bitstr(" Forward");

}

}





/\*

\* Atmega16codes.c

\* Never falling Bot

\* Created: 08-07-2019 04:04:56 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

#define lcd PORTD

lcd\_4bitinit() {

lcd\_4bitcmd(0x02);

lcd\_4bitcmd(0x28);

lcd\_4bitcmd(0x0c);

}

lcd\_4bitstr(char \*str) {

int d=0;

while(str[d]!='\0')

{

lcd\_4bitdata(str[d]);

lcd\_4bitcmd(0x06);

d++;

}

}

lcd\_4bitdata(int x) {

lcd=(x&0xf0)+0x05;

\_delay\_ms(2);

lcd=lcd-0x04;

lcd=((x<<4)&0xf0)+0x05;

\_delay\_ms(2);

lcd=lcd-0x04;

}

lcd\_4bitcmd(int x) {

lcd=(x&0xf0)+0x04;

\_delay\_ms(2);

lcd=lcd-0x04;

lcd=((x<<4)&0xf0)+0x04;

\_delay\_ms(2);

lcd=lcd-0x04;

}

lcd\_4bitnum(int n) {

int d;

if(n==0){lcd\_4bitdata(48);}

lcd\_4bitcmd(0x04);

while(n!=0)

{

d=n%10;

lcd\_4bitdata(d+48);

n=n/10;

}

}

int main(void)

{

DDRC = DDRD = 0xff;

DDRA = 0xfe;

lcd\_4bitinit();

while(1)

{

lcd\_4bitcmd(0x81);

lcd\_4bitstr(" N\_F");

while(PINA == 0x00)

{

PORTC = 0x00;

lcd\_4bitcmd(0x01); \_delay\_ms(10);

lcd\_4bitcmd(0xc3);

lcd\_4bitstr(" Edge Stop");

\_delay\_ms(500); //stop

PORTC = 0x06;

lcd\_4bitcmd(0x01); \_delay\_ms(10);

lcd\_4bitcmd(0xc3);

lcd\_4bitstr(" Backward");

\_delay\_ms(1000); //Backward

PORTC = 0x08;

lcd\_4bitcmd(0x01); \_delay\_ms(10);

lcd\_4bitcmd(0xc3);

lcd\_4bitstr(" Right turn");

\_delay\_ms(1000); //Right turn

lcd\_4bitcmd(0x01); \_delay\_ms(10);

}

PORTC = 0x09; //forward

lcd\_4bitcmd(0xc3);

lcd\_4bitstr("Forward");

}

}



/\*

\* Atmega16codes.c

\* LM35 interfacing

\* Created: 12-07-2019 03:44:40 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

#define lcd PORTB

lcd\_4bitinit()

{

lcd\_4bitcmd(0x02);

lcd\_4bitcmd(0x28);

lcd\_4bitcmd(0x0c);

}

lcd\_4bitstr(char \*str) {

int d=0;

while(str[d]!='\0')

{

lcd\_4bitdata(str[d]);

lcd\_4bitcmd(0x06);

d++;

}

}

lcd\_4bitdata(int x) {

lcd=(x&0xf0)+0x05;

\_delay\_ms(2);

lcd=lcd-0x04;

lcd=((x<<4)&0xf0)+0x05;

\_delay\_ms(2);

lcd=lcd-0x04;

}

lcd\_4bitcmd(int x) {

lcd=(x&0xf0)+0x04;

\_delay\_ms(2);

lcd=lcd-0x04;

lcd=((x<<4)&0xf0)+0x04;

\_delay\_ms(2);

lcd=lcd-0x04;

}

lcd\_4bitnum(int n) {

int d;

if(n==0){lcd\_4bitdata(48);}

lcd\_4bitcmd(0x04);

while(n!=0)

{

d=n%10;

lcd\_4bitdata(d+48);

n=n/10;

}

}

int ADC\_read( int chn)

{

ADMUX = 0x40 + chn ;

ADCSRA = 0xc7;

while ((ADCSRA & (0x10)) == 0){ }

return ADC;

}

int main(void)

{

DDRB = 0xff;

lcd\_4bitinit();

lcd\_4bitcmd(0x01); \_delay\_ms(10);

lcd\_4bitcmd(0x80);

lcd\_4bitstr(" Analog :");

lcd\_4bitcmd(0xc0);

lcd\_4bitstr(" Temp V :");

int n = 0;

while(1)

{

n = ADC\_read(0);

int t = n/2 - 1;

if (n <100)

{

lcd\_4bitcmd(0x89);

lcd\_4bitstr(" ");

lcd\_4bitcmd(0xc9);

lcd\_4bitstr(" ");

}

lcd\_4bitcmd(0x8b);

lcd\_4bitnum(n);

lcd\_4bitcmd(0xcb);

lcd\_4bitnum(t);

}

}





/\*

\* Atmega16codes.c

\* Linefollower Bot

\* Created: 12-07-2019 05:07:12 PM

\* Author: Gulshan Johara

\*/

#include <avr/io.h>

#include <util/delay.h>

#define lcd PORTD

lcd\_4bitinit() {

lcd\_4bitcmd(0x02);

lcd\_4bitcmd(0x28);

lcd\_4bitcmd(0x0c);

}

lcd\_4bitstr(char \*str) {

int d=0;

while(str[d]!='\0')

{

lcd\_4bitdata(str[d]);

lcd\_4bitcmd(0x06);

d++;

}

}

lcd\_4bitdata(int x) {

lcd=(x&0xf0)+0x05;

\_delay\_ms(2);

lcd=lcd-0x04;

lcd=((x<<4)&0xf0)+0x05;

\_delay\_ms(2);

lcd=lcd-0x04;

}

lcd\_4bitcmd(int x) {

lcd=(x&0xf0)+0x04;

\_delay\_ms(2);

lcd=lcd-0x04;

lcd=((x<<4)&0xf0)+0x04;

\_delay\_ms(2);

lcd=lcd-0x04;

}

lcd\_4bitnum(int n) {

int d;

if(n==0){lcd\_4bitdata(48);}

lcd\_4bitcmd(0x04);

while(n!=0)

{

d=n%10;

lcd\_4bitdata(d+48);

n=n/10;

}

}

int main(void)

{

DDRC = DDRD = 0xff;

DDRA = 0xfe;

lcd\_4bitinit();

lcd\_4bitcmd(0x01); \_delay\_ms(10);

lcd\_4bitcmd(0x81); lcd\_4bitstr(" LineFollower");

while(1)

{

if( PINA == 0b00000011)

{

PORTC = 0x09;

lcd\_4bitcmd(0xc1);

lcd\_4bitstr(" F"); //Forward

// \_delay\_ms(100);

}

else if (PINA == 0b00000010)

{

PORTC = 0x01;

lcd\_4bitcmd(0xc1);

lcd\_4bitstr(" L"); //Left

// \_delay\_ms(100);

}

else if (PINA == 0b00000001)

{

PORTC = 0x08;

lcd\_4bitcmd(0xc1);

lcd\_4bitstr(" R"); //Right

// \_delay\_ms(100);

}

else if(PINA == 0b00000000)

{

PORTC = 0x00;

lcd\_4bitcmd(0xc1);

lcd\_4bitstr(" S"); //Stop

// \_delay\_ms(100);

}

}

}



Receiver End (BOT section) + HC 05 Module



Transmitter End ( with HC 05 / Mobile )

/\*

\* RXTX.c

\*

\* Created: 24-06-2019 10:41:28 AM

\* Author: gulshan

\*/

#define F\_CPU 8000000

#include <avr/io.h>

#include <util/delay.h>

USART\_init()

{

UCSRB = 0x18;

UCSRC = 0x86;

UBRRL = 51;

UBRRH = 0;

}

char USART\_TX( char sead)

{

while ((UCSRA & (0x20)) == 0){}

UDR = sead;

}

char USART\_RX()

{

while ((UCSRA & (0x80)) == 0){}

return UDR;

}

int main(void)

{

DDRC = 0xff;

USART\_init();

while(1)

{

char a = USART\_RX();

USART\_TX(a);

if ( a=='F')

{

PORTC = 0b00001001;

\_delay\_ms(100);

}

else if (a == 'B')

{

PORTC = 0b00000110;

\_delay\_ms(100);

}

else if (a == 'R')

{

PORTC = 0b00001000;

\_delay\_ms(100);

}

else if (a == 'L')

{

PORTC = 0b00000001;

\_delay\_ms(100);

}

else

{

PORTC = 0b00000000;

}

}

}