

```

/*
 * Debug.c
 *
 * Created: 2019-04-04 16:42:49
 * Author : fr128401-s
 */
#define F_CPU 8000000UL

#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include <avr/eeprom.h>

#define LED1 0
#define BUSY 7
#define paddleStartX 10

void turnOffDisplay(void);
void turnOnDisplay(void);
void enableLCD(void);
void writeSomething(void);
void setScreen1(char);
void setScreen2(char);
void drawScreen1(char);
void drawScreen2(char);
void setPixelOn(char, char);
void clearDisplay(void);
void setPixelOff(char, char);
void draw_left_paddle(char);
void draw_right_paddle(char);
void draw_ball(char, char);
unsigned char getCurrent(void);
unsigned char getCurrent2(void);
void init_adc(void);
void start_adc(void);
volatile uint16_t joystick0;
volatile uint16_t joystick1;
void init_display(void);
void clear_left_paddle(void);
void wait(void);
void erase_ball(char, char);
char opposite(char);
void clear_right_paddle(void);
void set_and_draw(char, char, char);
void draw_line(char);

void draw_A(char, char);
void draw_B(char, char);
void draw_C(char, char);
void draw_D(char, char);
void draw_E(char, char);
void draw_F(char, char);
void draw_G(char, char);
void draw_H(char, char);

```

```

void draw_I(char, char);
void draw_J(char, char);
void draw_K(char, char);
void draw_L(char, char);
void draw_M(char, char);
void draw_N(char, char);
void draw_O(char, char);
void draw_P(char, char);
void draw_Q(char, char);
void draw_R(char, char);
void draw_S(char, char);
void draw_T(char, char);
void draw_U(char, char);
void draw_V(char, char);
void draw_W(char, char);
void draw_X(char, char);
void draw_Y(char, char);
void draw_Z(char, char);

void draw_0(char, char);
void draw_1(char, char);
void draw_2(char, char);
void draw_3(char, char);
void draw_4(char, char);
void draw_5(char, char);
void draw_6(char, char);
void draw_7(char, char);
void draw_8(char, char);
void draw_9(char, char);

void draw_square(char, char);
void set_and_draw1(char, char, char);
void erase_square(char, char);
void erase_square_down(void);
void erase_square_up(void);
void draw_square_up(void);
void draw_square_down(void);
void draw_letter2(char, char, char);
void draw_number(char, char, char);

char blstate;

volatile uint16_t adc_value;

int main(void){

    init_display();

    DDRC^=(1<<0);

    init_adc();

    sei();
    start_adc();

```

```

while(1){

clearDisplay();

draw_P(2,5);
draw_L(2,12);
draw_A(2,19);
draw_Y(2,26);

draw_H(5,5);
draw_I(5,12);
draw_S(5,19);
draw_T(5,26);
draw_O(5,33);
draw_R(5,40);
draw_Y(5,47);

draw_square_up();

char state = 0;

b1state = 1;
while(b1state){
    b1state = (PINB & 0b10000000)>>1;
    if(state == 0 && joystick1>700){
        erase_square_up();
        draw_square_down();
        state = 1;
    }else if(state == 1 && joystick1 <300){
        erase_square_down();
        draw_square_up();
        state = 0;
    }
}

if(state){

clearDisplay();
draw_letter2(0,0,EEPROM_READ_BYTE((const uint8_t *) 2));
draw_letter2(0,7,EEPROM_READ_BYTE((const uint8_t *) 3));
draw_letter2(0,14,EEPROM_READ_BYTE((const uint8_t *) 4));

draw_letter2(0,28,EEPROM_READ_BYTE((const uint8_t *) 5));
draw_letter2(0,35,EEPROM_READ_BYTE((const uint8_t *) 6));
draw_letter2(0,42,EEPROM_READ_BYTE((const uint8_t *) 7));

set_and_draw(1,21,0b00001000);
set_and_draw(1,22,0b00001000);
set_and_draw(1,23,0b00001000);
set_and_draw(1,24,0b00001000);

draw_number(1, 7,EEPROM_READ_BYTE((const uint8_t *) 0));
draw_number(1, 33,EEPROM_READ_BYTE((const uint8_t *) 1));

```

```

for (char i =0;i<64;i++){
set_and_draw(2,i,0b00001000);
}

draw_letter2(3,0,EEPROM_READ_BYTE((const uint8_t *) 10));
draw_letter2(3,7,EEPROM_READ_BYTE((const uint8_t *) 11));
draw_letter2(3,14,EEPROM_READ_BYTE((const uint8_t *) 12));

draw_letter2(3,28,EEPROM_READ_BYTE((const uint8_t *) 13));
draw_letter2(3,35,EEPROM_READ_BYTE((const uint8_t *) 14));
draw_letter2(3,42,EEPROM_READ_BYTE((const uint8_t *) 15));

set_and_draw(4,21,0b00001000);
set_and_draw(4,22,0b00001000);
set_and_draw(4,23,0b00001000);
set_and_draw(4,24,0b00001000);

draw_number(4, 7,EEPROM_READ_BYTE((const uint8_t *) 8));
draw_number(4, 33,EEPROM_READ_BYTE((const uint8_t *) 9));

for (char i =0;i<64;i++){
    set_and_draw(5,i,0b00001000);
}

draw_letter2(6,0,EEPROM_READ_BYTE((const uint8_t *) 18));
draw_letter2(6,7,EEPROM_READ_BYTE((const uint8_t *) 19));
draw_letter2(6,14,EEPROM_READ_BYTE((const uint8_t *) 20));

draw_letter2(6,28,EEPROM_READ_BYTE((const uint8_t *) 21));
draw_letter2(6,35,EEPROM_READ_BYTE((const uint8_t *) 22));
draw_letter2(6,42,EEPROM_READ_BYTE((const uint8_t *) 23));

set_and_draw(7,21,0b00001000);
set_and_draw(7,22,0b00001000);
set_and_draw(7,23,0b00001000);
set_and_draw(7,24,0b00001000);

draw_number(7, 7,EEPROM_READ_BYTE((const uint8_t *) 16));
draw_number(7, 33,EEPROM_READ_BYTE((const uint8_t *) 17));

b1state = 1;
while(b1state){
    b1state = (PINB & 0b10000000)>>1;
}

if(!state){
clearDisplay();
draw_P(0,0);
draw_L(0,7);
draw_A(0,14);
draw_Y(0,21);
draw_E(0,28);
draw_R(0,35);
}

```

```

draw_1(0,46);

draw_P(4,0);
draw_L(4,7);
draw_A(4,14);
draw_Y(4,21);
draw_E(4,28);
draw_R(4,35);
draw_2(4,46);

draw_A(2,12);
draw_A(2,26);
draw_A(2,40);

draw_A(6,12);
draw_A(6,26);
draw_A(6,40);

char player1_name[3];
char player2_name[3];
char currentPos = 0;
    draw_line(currentPos);
char currentLetter = 0;
while(currentPos !=6 || b1state){
    b1state = (PINB & 0b10000000)>>1;
    if(joystick1 > 700){

        if(currentLetter <25){
            currentLetter = currentLetter +1;
        }else{
            currentLetter = 0;
        }

        draw_letter(currentPos, currentLetter);
    }

    if(!b1state){
        if(currentPos <3){
            player1_name[currentPos] = currentLetter;
        }else{
            player2_name[currentPos-3] =
currentLetter;
        }

        currentLetter = 0;
        currentPos = currentPos +1;
        draw_line(currentPos);
    }

    _delay_ms(100);

```

```

}

clearDisplay();

setScreen1(0b01000000);
setScreen1(0b10111000);

for(char j = 0; j<124;j++){
    setPixelOn(7,j);
}

setScreen1(0b01000000);
setScreen1(0b10111000);

for(char j = 0; j<124;j++){
    setPixelOn(57,j);
}

draw_left_paddle(paddleStartX);
draw_right_paddle(paddleStartX);

draw_letter2(0,5,player1_name[0]);
draw_letter2(0,11,player1_name[1]);
draw_letter2(0,17,player1_name[2]);

set_and_draw(0,25,0b00001000);
set_and_draw(0,26,0b00001000);
set_and_draw(0,27,0b00001000);
set_and_draw(0,28,0b00001000);

draw_letter2(0,32,player2_name[0]);
draw_letter2(0,39,player2_name[1]);
draw_letter2(0,46,player2_name[2]);

char p1Pos = paddleStartX;
char p2Pos = paddleStartX;
char ballX = 32;
char ballY = 64;
char ballDirection = 0;
char p1Points = 0;
char p2Points = 0;

```

```

while(p1Points<3 && p2Points <3){
    /*****UPDATE JOYSTICKS*****/

    if(joystick1<300 && p1Pos > 4){
        p1Pos = p1Pos -3;
    }
    if(joystick1>700 && p1Pos <46){
        p1Pos = p1Pos +3;
    }
    if(joystick0<300 && p2Pos > 4){
        p2Pos = p2Pos -3;
    }
    if(joystick0>700 && p2Pos <46){
        p2Pos = p2Pos +3;
    }

    clear_left_paddle();
    draw_left_paddle(p1Pos);

    clear_right_paddle();
    draw_right_paddle(p2Pos);

    /*****ERASE OLD BALL*****/
    erase_ball(ballX,ballY);

    /*****LEFT PADDLE*****/

    if(ballY < 6 && p1Pos < ballX && p1Pos+6 > ballX){
        ballDirection = 2;
    }else if(ballY < 6 && p1Pos +5 < ballX && p1Pos+11 >
ballX){
        ballDirection = 1;
    }else if(ballY < 6 && p1Pos +10 < ballX && p1Pos+15
> ballX){
        ballDirection = 4;
    }else if(ballY < 6){
        p2Points = p2Points +1;

        ballX = 32;
        ballY = 64;
    }
}

```

```

/*****RIGHT PADDLE*****/

    if(ballY > 122 && p2Pos < ballX && p2Pos+6 > ballX){
        ballDirection = 3;
    }else if(ballY > 122 && p2Pos +5 < ballX && p2Pos+11
> ballX){
        ballDirection = 0;
    }else if(ballY > 122 && p2Pos +10 < ballX &&
p2Pos+15 > ballX){
        ballDirection = 5;
    }else if(ballY >122){
        p1Points = p1Points +1;

        ballX = 32;
        ballY = 64;
    }

/*****TOP & BOTTOM BOUNCE*****/
if(ballX<13){
    ballDirection = opposite(ballDirection);
}

if(ballX>49){
    ballDirection = opposite(ballDirection);
}

/*****MOVE BALL*****/
switch(ballDirection){
case 0 :
    ballY = ballY -10;
    break;
case 1:
    ballY = ballY +10;
    break;
case 2:
    ballY = ballY +7;
    ballX = ballX -7;
    break;
case 3:
    ballY = ballY -7;
    ballX = ballX -7;
    break;
case 4:
    ballY = ballY +7;
    ballX = ballX +7;
    break;
case 5:
    ballY = ballY -7;
    ballX = ballX +7;
    break;
}

```



```

        draw_ball(ballX,ballY);

    }

    clearDisplay();
    draw_letter2(1,0,player1_name[0]);
    draw_letter2(1,7,player1_name[1]);
    draw_letter2(1,14,player1_name[2]);

    set_and_draw(1,24,0b00001000);
    set_and_draw(1,25,0b00001000);
    set_and_draw(1,26,0b00001000);
    set_and_draw(1,27,0b00001000);

    draw_letter2(1,34,player2_name[0]);
    draw_letter2(1,41,player2_name[1]);
    draw_letter2(1,48,player2_name[2]);

    draw_number(3,7,p1Points);

    set_and_draw(3,24,0b00001000);
    set_and_draw(3,25,0b00001000);
    set_and_draw(3,26,0b00001000);
    set_and_draw(3,27,0b00001000);

    draw_number(3,41,p2Points);

    for (char i = 0;i<8;i++){
        eeprom_write_byte(16+i,eeprom_read_byte((const uint8_t *)
(8+i)));
    }

    for (char i = 0;i<8;i++){
        eeprom_write_byte(8+i,eeprom_read_byte((const uint8_t *) i));
    }
    eeprom_write_byte(0,p1Points);
    eeprom_write_byte(1,p2Points);
    eeprom_write_byte(2,player1_name[0]);
    eeprom_write_byte(3,player1_name[1]);
    eeprom_write_byte(4,player1_name[2]);
    eeprom_write_byte(5,player2_name[0]);
    eeprom_write_byte(6,player2_name[1]);
    eeprom_write_byte(7,player2_name[2]);

    _delay_ms(4000);

    }
}

```

```
}
```

```
//GAME FUNCTIONS
```

```
char opposite(char c){  
    switch(c){  
        case 2:  
            return 4    ;  
            break;  
        case 3:  
            return 5;  
            break;  
        case 4:  
            return 2;  
            break;  
        case 5:  
            return 3;  
            break;  
    }  
  
    return 0;  
}
```

```
}
```

```
//DISPLAY FUNCTIONS
```

```
void drawScreen1(char c){  
  
    //sätt di till 1  
    PORTB |= (1<<4);  
  
    //fyll i charens värde i PORTD  
    PORTD = c;  
  
    //enable  
    enableLCD();  
  
    //återgå till ursprungsläget för di  
    PORTB &= 0b11101111;  
}
```

```
void drawScreen2(char c){
```

```
    //byt chip till 2  
    PORTB |= (1<<1);
```

```

    PORTB &= 0b11111110;

    //sätt di till 1
    PORTB |= (1<<4);

    //fyll i charens värde i PORTD
    PORTD = c;

    //enable
    enableLCD();

    //återgå till ursprungsläget för di
    PORTB &= 0b11101111;

    //Återgå till ursprungsläge chip
    PORTB |= (1<<0);
    PORTB &= 0b11111101;

}

void setScreen1(char c){

    //fyll i charens värde i PORTD
    PORTD = c;

    //enable
    enableLCD();
}

void setScreen2(char c){

    //byt chip till 2
    PORTB |= (1<<1);
    PORTB &=~(1<<0);

    //fyll i charens värde i PORTD
    PORTD = c;

    //enable
    enableLCD();

    //Återgå till ursprungsläge chip
    PORTB |= (1<<0);
    PORTB &= 0b11111101;
}

void turnOnDisplay(){

```

```

        setScreen1(0b00111111);
        setScreen2(0b00111111);
    }

void turnOffDisplay(){
    setScreen1(0b00111110);
    setScreen2(0b00111110);
}

void setPixelOn(char x, char y){

    char tempX = x/8;
    char tempP = x % 8;
    char tempPixel = (1<<tempP);

    char tempY = y;

    if(y<64){
        tempX |= 0b10111000;

        tempY |= 0b01000000;

        setScreen1(tempX);
        setScreen1(tempY);

        // unsigned char curr = getCurrent();

        // tempPixel |= curr;

        setScreen1(tempX);
        setScreen1(tempY);

        drawScreen1(tempPixel);
    }else{

        tempX |= 0b10111000;

        tempY = tempY - 64;

        tempY |= 0b01000000;
        setScreen2(tempX);
        setScreen2(tempY);

        // unsigned char curr2 = getCurrent2();
        // tempPixel |= curr2;
        setScreen2(tempX);
        setScreen2(tempY);
        drawScreen2(tempPixel);
    }
}

unsigned char getCurrent(){

    DDRD = 0b00000000;
    PORTB |= (1<<3);
    PORTB |= (1<<4);
}

```

```

    enableLCD();

    unsigned char current = PIND;

    DDRD = 0b11111111;
    PORTB &=~(1<<3);
    PORTB &=~(1<<4);

    return current;
}

unsigned char getCurrent2 (){

    //byt chip till 2
    PORTB |= (1<<1);
    PORTB &=~(1<<0);

    DDRD = 0b00000000;
    PORTB |= (1<<3);
    PORTB |= (1<<4);

    enableLCD();

    unsigned char current2 = PIND;

    DDRD = 0b11111111;
    PORTB &=~(1<<3);
    PORTB &=~(1<<4);

    //Återgå till ursprungsläge chip
    PORTB |= (1<<0);
    PORTB &= 0b11111101;

    return current2;
}

void setPixelOff(char x, char y){

    char tempX = x/8;
    char tempP = x % 8;
    char tempPixel = ~(1<<tempP);

    char tempY = y;

    if(y<64){
        tempX |= 0b10111000;
    }
}

```

```

        tempY |= 0b01000000;

        setScreen1(tempX);
        setScreen1(tempY);

        unsigned char curr = getCurrent();

        tempPixel &= curr;

        setScreen1(tempX);
        setScreen1(tempY);

        drawScreen1(tempPixel);
    }else{

        tempX |= 0b10111000;

        tempY = tempY - 64;

        tempY |= 0b01000000;
        setScreen2(tempX);
        setScreen2(tempY);
        unsigned char curr2 = getCurrent2();
        tempPixel &= curr2;
        setScreen2(tempX);
        setScreen2(tempY);
        drawScreen2(tempPixel);
    }
}

void clearDisplay(){
    for(int j = 0; j < 8 ; j++){
        volatile char c = 184 + j;

        setScreen1(c);

        for(int i = 0; i <64;i++){
            drawScreen1(0);
        }
    }

    for(int j = 0; j < 8 ; j++){
        volatile char c = 184 + j;

        setScreen2(c);

        for(int i = 0; i <64;i++){
            drawScreen2(0);
        }
    }
}

```

```

    }
}

void clear_left_paddle(){
    for(int j = 0; j < 8 ; j++){
        volatile char c = 184 + j;
        setScreen1(c);
        setScreen1(0b01000001);
        drawScreen1(0);
    }
}

void clear_right_paddle(){
    for(int j = 0; j < 8 ; j++){
        volatile char c = 184 + j;
        setScreen2(c);
        setScreen2(0b01111111);
        drawScreen2(0);
    }
}

void init_display(){
    DDRB=0b00111111;
    DDRD=0b11111111;
    PORTB = 0b00100101;
    turnOnDisplay();

    setScreen1(0b10111000);
    setScreen1(0b01000000);

    setScreen2(0b10111000);
    setScreen2(0b01000000);
}

void wait(){
    //Ändra R/W och Inåtrikta D

```

```

    DDRD = 0;
    PORTB |= (1<<3);

    enableLCD();
    while(PIND & (1<<BUSY) ){

        PORTC |= (1 << LED1);

        //          enableLCD();
    }

    //byt chip till 2
    PORTB |= (1<<1);
    PORTB &=~(1<<0);

    enableLCD();

    while(PIND & (1<<BUSY) ){

        PORTC |= (1 << LED1);

        //          enableLCD();
    }

    PORTC &= ~(1 << LED1);

    //Återgå till ursprungsläge chip
    PORTB |= (1<<0);
    PORTB &= 0b11111101;

    DDRD = 0b11111111;
    PORTB &=~(1<<3);
}

void enableLCD(){

    _delay_ms(0.5);

    PORTB &=~(1<<5);

    PORTB |= (1<<5);

    _delay_ms(0.5);
}

//DRAWING FUNCTIONS

void draw_left_paddle(char x){

```



```

char xx = x%8;
setScreen1(0b01000001);
setScreen1(0b10111000 | x/8);

switch(xx){
case 0:

drawScreen1(0b11111111);
setScreen1(0b01000001);
setScreen1((0b10111000 | x/8) +1);
drawScreen1(0b11111111);
break;

case 1:
drawScreen1(0b11111110);
setScreen1(0b01000001);
setScreen1((0b10111000 | x/8) +1);
drawScreen1(0b11111111);
setScreen1(0b01000001);
setScreen1((0b10111000 | x/8) +2);
drawScreen1(0b00000001);
break;

case 2:
drawScreen1(0b11111100);
setScreen1(0b01000001);
setScreen1((0b10111000 | x/8) +1);
drawScreen1(0b11111111);
setScreen1(0b01000001);
setScreen1((0b10111000 | x/8) +2);
drawScreen1(0b00000011);
break;

case 3:
drawScreen1(0b11111000);
setScreen1(0b01000001);
setScreen1((0b10111000 | x/8) +1);
drawScreen1(0b11111111);
setScreen1(0b01000001);
setScreen1((0b10111000 | x/8) +2);
drawScreen1(0b00000111);
break;

case 4:
drawScreen1(0b11110000);
setScreen1(0b01000001);
setScreen1((0b10111000 | x/8) +1);
drawScreen1(0b11111111);
setScreen1(0b01000001);
setScreen1((0b10111000 | x/8) +2);
drawScreen1(0b00001111);
break;
case 5:
drawScreen1(0b11100000);
setScreen1(0b01000001);
setScreen1((0b10111000 | x/8) +1);
drawScreen1(0b11111111);
setScreen1(0b01000001);

```

```

    setScreen1((0b10111000 | x/8) +2);
    drawScreen1(0b00011111);
    break;
    case 6:
    drawScreen1(0b11000000);
    setScreen1(0b01000001);
    setScreen1((0b10111000 | x/8) +1);
    drawScreen1(0b11111111);
    setScreen1(0b01000001);
    setScreen1((0b10111000 | x/8) +2);
    drawScreen1(0b00011111);
    break;
    case 7:
    drawScreen1(0b10000000);
    setScreen1(0b01000001);
    setScreen1((0b10111000 | x/8) +1);
    drawScreen1(0b11111111);
    setScreen1(0b01000001);
    setScreen1((0b10111000 | x/8) +2);
    drawScreen1(0b01111111);
    break;
    }

    //printa paddeln
    //for(char i = x; i<x+16; i++){

//          setPixelOn(i,0);
//      }
}

void draw_right_paddle(char x){

    char xx = x%8;
    setScreen2(0b01111111);
    setScreen2(0b10111000 | x/8);

    switch(xx){
        case 0:

            drawScreen2(0b11111111);
            setScreen2(0b01111111);
            setScreen2((0b10111000 | x/8) +1);
            drawScreen2(0b11111111);
            break;

        case 1:
            drawScreen2(0b11111110);
            setScreen2(0b01111111);
            setScreen2((0b10111000 | x/8) +1);
            drawScreen2(0b11111111);
            setScreen2(0b01111111);
            setScreen2((0b10111000 | x/8) +2);
            drawScreen2(0b00000001);
            break;
    }
}

```

```
case 2:
drawScreen2(0b11111100);
setScreen2(0b01111111);
setScreen2((0b10111000 | x/8) +1);
drawScreen2(0b11111111);
setScreen2(0b01111111);
setScreen2((0b10111000 | x/8) +2);
drawScreen2(0b00000011);
break;
```

```
case 3:
drawScreen2(0b11111000);
setScreen2(0b01111111);
setScreen2((0b10111000 | x/8) +1);
drawScreen2(0b11111111);
setScreen2(0b01111111);
setScreen2((0b10111000 | x/8) +2);
drawScreen2(0b00000111);
break;
```

```
case 4:
drawScreen2(0b11110000);
setScreen2(0b01111111);
setScreen2((0b10111000 | x/8) +1);
drawScreen2(0b11111111);
setScreen2(0b01111111);
setScreen2((0b10111000 | x/8) +2);
drawScreen2(0b00001111);
break;
```

```
case 5:
drawScreen2(0b11100000);
setScreen2(0b01111111);
setScreen2((0b10111000 | x/8) +1);
drawScreen2(0b11111111);
setScreen2(0b01111111);
setScreen2((0b10111000 | x/8) +2);
drawScreen2(0b00011111);
break;
```

```
case 6:
drawScreen2(0b11000000);
setScreen2(0b01111111);
setScreen2((0b10111000 | x/8) +1);
drawScreen2(0b11111111);
setScreen2(0b01111111);
setScreen2((0b10111000 | x/8) +2);
drawScreen2(0b00111111);
break;
```

```
case 7:
drawScreen2(0b10000000);
setScreen2(0b01111111);
setScreen2((0b10111000 | x/8) +1);
drawScreen2(0b11111111);
setScreen2(0b01111111);
setScreen2((0b10111000 | x/8) +2);
drawScreen2(0b01111111);
break;
```

```
}
```

```

//printa paddeln
//for(char i = x; i<x+16; i++){
//
//          //setPixelOn(i,125);
//}
}

```

```

void draw_ball(char x, char y){
//          // printa bollen
//

```

```

switch(x%8){

case 0:
if(y<64){
set_and_draw1(x/8,y,0b00000011);
set_and_draw1(x/8,y+1,0b00000011);
}else{
set_and_draw(x/8,y-64,0b00000011);
set_and_draw(x/8,y-63,0b00000011);
}
break;
case 1:
if(y<64){
set_and_draw1(x/8,y,0b00000110);
set_and_draw1(x/8,y+1,0b00000110);
}else{
set_and_draw(x/8,y-64,0b00000110);
set_and_draw(x/8,y-63,0b00000110);
}

break;
case 2:
if(y<64){
set_and_draw1(x/8,y,0b00001100);
set_and_draw1(x/8,y+1,0b00001100);
}else{
set_and_draw(x/8,y-64,0b00001100);
set_and_draw(x/8,y-63,0b00001100);
}

break;
case 3:
if(y<64){
set_and_draw1(x/8,y,0b00011000);

```

```

        set_and_draw1(x/8,y+1,0b00011000);
    }else{
        set_and_draw(x/8,y-64,0b00011000);
        set_and_draw(x/8,y-63,0b00011000);
    }

    break;
    case 4:
    if(y<64){
        set_and_draw1(x/8,y,0b00110000);
        set_and_draw1(x/8,y+1,0b00110000);
    }else{
        set_and_draw(x/8,y-64,0b00110000);
        set_and_draw(x/8,y-63,0b00110000);
    }

    break;
    case 5:
    if(y<64){
        set_and_draw1(x/8,y,0b01100000);
        set_and_draw1(x/8,y+1,0b01100000);
    }else{
        set_and_draw(x/8,y-64,0b01100000);
        set_and_draw(x/8,y-63,0b01100000);
    }

    break;
    case 6:
    if(y<64){
        set_and_draw1(x/8,y,0b11000000);
        set_and_draw1(x/8,y+1,0b11000000);
    }else{
        set_and_draw(x/8,y-64,0b11000000);
        set_and_draw(x/8,y-63,0b11000000);
    }

    break;
    case 7:
    if(y<64){
        set_and_draw1(x/8,y,0b10000000);
        set_and_draw1(x/8,y+1,0b10000000);
        set_and_draw1(x/8+1,y,0b00000001);
        set_and_draw1(x/8+1,y+1,0b00000001);
    }else{
        set_and_draw(x/8,y-64,0b10000000);
        set_and_draw(x/8,y-63,0b10000000);
        set_and_draw(x/8+1,y-64,0b00000001);
        set_and_draw(x/8+1,y-63,0b00000001);
    }
    break;
}

}

}

void erase_ball(char x, char y){

```

```

if(y<64){
    set_and_draw1(x/8,y,0b00000000);
    set_and_draw1(x/8,y+1,0b00000000);
    if(x%8 == 7){
        set_and_draw1(x/8+1,y,0b00000000);
        set_and_draw1(x/8+1,y+1,0b00000000);
    }
    }else{
        set_and_draw(x/8,y-64,0b00000000);
        set_and_draw(x/8,y-63,0b00000000);
    if(x%8 == 7){
        set_and_draw(x/8+1,y-64,0b00000000);
        set_and_draw(x/8+1,y-63,0b00000000);
    }
    }
}

```

```

//AD CONVERTER FUNCTIONS

```

```

void init_adc(){
    ADMUX |= (1<<REFS0);
    ADCSRA |=(1<<ADEN)|(1<<ADIE)|(1<<ADPS2)|(1<<ADPS0);
}

```

```

void start_adc(){
    ADCSRA |= (1<<ADSC);
}

```

```

//void init_led(){
//    DDRC |= (1<<LED1);
//}

```

```

ISR(ADC_vect) {
    if(ADMUX & (1<<MUX0)){
        joystick1 = ADC;
    }else{
        joystick0 = ADC;
    }
    ADMUX ^= (1<<MUX0);
}

```

```

        start_adc();
    }

    /*Rita alla tecken*/
    void erase_square_down(){
        erase_square(5,40);
    }
    void erase_square_up(){
        erase_square(2,40);
    }
    void draw_square_down(){
        draw_square(5,40);
    }
    void draw_square_up(){
        draw_square(2,40);
    }

    void erase_square(char x, char y){

        set_and_draw1(x,y, 0b00000000);
        set_and_draw1(x,y+1, 0b00000000);
        set_and_draw1(x,y+2, 0b00000000);
        set_and_draw1(x,y+3, 0b00000000);
        set_and_draw1(x,y+4, 0b00000000);

    }

    void draw_square(char x, char y){

        set_and_draw1(x,y, 0b01111111);
        set_and_draw1(x,y+1, 0b01111111);
        set_and_draw1(x,y+2, 0b01111111);
        set_and_draw1(x,y+3, 0b01111111);
        set_and_draw1(x,y+4, 0b01111111);

    }

    void set_and_draw(char x, char y, char c){
        setScreen2(0b01000000 | y);
        setScreen2(0b10111000 | x);

        drawScreen2(c);
    }

    void set_and_draw1(char x, char y, char c){
        setScreen1(0b01000000 | y);
        setScreen1(0b10111000 | x);

        drawScreen1(c);
    }

    /****** LETTERS *****/

```

```
void draw_line(char pos){
    switch(pos){
        case 0:
            set_and_draw(3,12,1);
            set_and_draw(3,13,1);
            set_and_draw(3,14,1);
            set_and_draw(3,15,1);
            set_and_draw(3,16,1);
            break;
        case 1:
            set_and_draw(3,12,0);
            set_and_draw(3,13,0);
            set_and_draw(3,14,0);
            set_and_draw(3,15,0);
            set_and_draw(3,16,0);
            set_and_draw(3,26,1);
            set_and_draw(3,27,1);
            set_and_draw(3,28,1);
            set_and_draw(3,29,1);
            set_and_draw(3,30,1);
            break;
        case 2:
            set_and_draw(3,26,0);
            set_and_draw(3,27,0);
            set_and_draw(3,28,0);
            set_and_draw(3,29,0);
            set_and_draw(3,30,0);
            set_and_draw(3,40,1);
            set_and_draw(3,41,1);
            set_and_draw(3,42,1);
            set_and_draw(3,43,1);
            set_and_draw(3,44,1);
            break;
        case 3:
            set_and_draw(3,40,0);
            set_and_draw(3,41,0);
            set_and_draw(3,42,0);
            set_and_draw(3,43,0);
            set_and_draw(3,44,0);
            set_and_draw(7,12,1);
            set_and_draw(7,13,1);
            set_and_draw(7,14,1);
            set_and_draw(7,15,1);
            set_and_draw(7,16,1);
            break;
        case 4:
            set_and_draw(7,12,0);
            set_and_draw(7,13,0);
            set_and_draw(7,14,0);
            set_and_draw(7,15,0);
            set_and_draw(7,16,0);
            set_and_draw(7,26,1);
            set_and_draw(7,27,1);
            set_and_draw(7,28,1);
            set_and_draw(7,29,1);
            set_and_draw(7,30,1);
            break;
        case 5:
```



```
set_and_draw(7,26,0);
set_and_draw(7,27,0);
set_and_draw(7,28,0);
set_and_draw(7,29,0);
set_and_draw(7,30,0);
set_and_draw(7,40,1);
set_and_draw(7,41,1);
set_and_draw(7,42,1);
set_and_draw(7,43,1);
set_and_draw(7,44,1);
break;
```

```
}
```

```
}
```

```
void draw_letter2(char xx, char yy, char currLetter){
```

```
switch (currLetter){
    case 0:
        draw_A(xx,yy);
        break;
    case 1:
        draw_B(xx,yy);
        break;
    case 2:
        draw_C(xx,yy);
        break;
    case 3:
        draw_D(xx,yy);
        break;
    case 4:
        draw_E(xx,yy);
        break;
    case 5:
        draw_F(xx,yy);
        break;
    case 6:
        draw_G(xx,yy);
        break;
    case 7:
        draw_H(xx,yy);
        break;
    case 8:
        draw_I(xx,yy);
        break;
    case 9:
        draw_J(xx,yy);
        break;
    case 10:
        draw_K(xx,yy);
        break;
    case 11:
        draw_L(xx,yy);
        break;
    case 12:
```

```

        draw_M(xx,yy);
        break;
        case 13:
        draw_N(xx,yy);
        break;
        case 14:
        draw_O(xx,yy);
        break;
        case 15:
        draw_P(xx,yy);
        break;
        case 16:
        draw_Q(xx,yy);
        break;
        case 17:
        draw_R(xx,yy);
        break;
        case 18:
        draw_S(xx,yy);
        break;
        case 19:
        draw_T(xx,yy);
        break;
        case 20:
        draw_U(xx,yy);
        break;
        case 21:
        draw_V(xx,yy);
        break;
        case 22:
        draw_W(xx,yy);
        break;
        case 23:
        draw_X(xx,yy);
        break;
        case 24:
        draw_Y(xx,yy);
        break;
        case 25:
        draw_Z(xx,yy);
        break;
    }
}

void draw_number(char x, char y, char nbr){
    switch(nbr){
        case 0:
        draw_0(x,y);
        break;
        case 1:
        draw_1(x,y);
        break;
        case 2:
        draw_2(x,y);
        break;

```

```
case 3:  
draw_3(x,y);  
break;  
case 4:  
draw_4(x,y);  
break;  
case 5:  
draw_5(x,y);  
break;  
case 6:  
draw_6(x,y);  
break;  
case 7:  
draw_7(x,y);  
break;  
case 8:  
draw_8(x,y);  
break;  
case 9:  
draw_9(x,y);  
break;
```

```
}
```

```
}
```

```
void draw_letter(char pos, char currLetter){  
    char xx = 0;  
    char yy = 0;  
  
    switch(pos){  
        case 0:  
            xx = 2;  
            yy = 12;  
            break;  
  
        case 1:  
            xx = 2;  
            yy = 26;  
            break;  
  
        case 2:  
            xx = 2;  
            yy = 40;  
            break;  
  
        case 3:  
            xx = 6;  
            yy = 12;  
            break;  
        case 4:  
            xx = 6;  
            yy = 26;  
            break;  
        case 5:  
            xx = 6;
```

```

        yy = 40;
        break;
    }

    draw_letter2(xx,yy, currLetter);
}

void draw_A(char x, char y){

    set_and_draw(x,y, 0b01111110);
    set_and_draw(x,y+1, 0b00001001);
    set_and_draw(x, y+2, 0b00001001);
    set_and_draw(x,y+3,0b00001001);
    set_and_draw(x,y+4,0b01111110);
}

void draw_B(char x, char y){

    set_and_draw(x,y, 0b01111111);
    set_and_draw(x,y+1, 0b01001001);
    set_and_draw(x,y+2, 0b01001001);
    set_and_draw(x,y+3,      0b01001001);
    set_and_draw(x,y+4,      0b00110110);
}

void draw_C(char x, char y){

    set_and_draw(x,y,      0b01111111);
    set_and_draw(x,y+1, 0b01000001);
    set_and_draw(x,y+2, 0b01000001);
    set_and_draw(x,y+3,      0b01000001);
    set_and_draw(x,y+4,      0b00100010);
}

void draw_D(char x, char y){

    set_and_draw(x,y,      0b01111111);
    set_and_draw(x,y+1, 0b01000001);
    set_and_draw(x,y+2, 0b01000001);
    set_and_draw(x,y+3,      0b01000001);
    set_and_draw(x,y+4,      0b00011100);
}

void draw_E(char x, char y){

    set_and_draw(x,y,      0b01111111);
    set_and_draw(x,y+1, 0b01001001);
    set_and_draw(x,y+2, 0b01001001);
    set_and_draw(x,y+3,      0b01001001);
    set_and_draw(x,y+4,      0b01000001);
}

void draw_F(char x, char y){

```

```

        set_and_draw(x,y,          0b01111111);
        set_and_draw(x,y+1, 0b00001001);
        set_and_draw(x,y+2, 0b00001001);
        set_and_draw(x,y+3,          0b00001001);
        set_and_draw(x,y+4,          0b00000001);
    }
void draw_G(char x, char y){

        set_and_draw(x,y,          0b01111110);
        set_and_draw(x,y+1, 0b01000001);
        set_and_draw(x,y+2, 0b01000001);
        set_and_draw(x,y+3,          0b01010001);
        set_and_draw(x,y+4,          0b00110010);
    }

void draw_H(char x, char y){

        set_and_draw(x,y,          0b01111111);
        set_and_draw(x,y+1,          0b00001000);
        set_and_draw(x, y+2, 0b00001000);
        set_and_draw(x,y+3, 0b00001000);
        set_and_draw(x,y+4, 0b01111111);
    }

void draw_I(char x, char y){

        set_and_draw(x,y,          0b00000000);
        set_and_draw(x,y+1, 0b01000001);
        set_and_draw(x,y+2, 0b01111111);
        set_and_draw(x,y+3,          0b01000001);
        set_and_draw(x,y+4, 0b00000000);
    }

void draw_J(char x, char y){

        set_and_draw(x,y,          0b00100000);
        set_and_draw(x,y+1, 0b01000000);
        set_and_draw(x,y+2, 0b01000001);
        set_and_draw(x,y+3,          0b00111111);
        set_and_draw(x,y+4,          0b00000001);
    }

void draw_K(char x, char y){

        set_and_draw(x,y,          0b01111111);
        set_and_draw(x,y+1, 0b00001000);
        set_and_draw(x,y+2, 0b00010100);
        set_and_draw(x,y+3,          0b00100010);
        set_and_draw(x,y+4,          0b01000001);
    }

void draw_L(char x, char y){

```

```

        set_and_draw(x,y,          0b01111111);
        set_and_draw(x,y+1,      0b01000000);
        set_and_draw(x, y+2, 0b01000000);
        set_and_draw(x,y+3, 0b01000000);
        set_and_draw(x,y+4, 0b01000000);
    }
    void draw_M(char x, char y){

        set_and_draw(x,y,          0b01111111);
        set_and_draw(x,y+1, 0b00000010);
        set_and_draw(x,y+2, 0b00001100);
        set_and_draw(x,y+3,          0b00000010);
        set_and_draw(x,y+4,          0b01111111);
    }

    void draw_N(char x, char y){

        set_and_draw(x,y,          0b01111111);
        set_and_draw(x,y+1, 0b00000100);
        set_and_draw(x,y+2, 0b00001000);
        set_and_draw(x,y+3,          0b00010000);
        set_and_draw(x,y+4,          0b01111111);
    }

    void draw_O(char x, char y){

        set_and_draw(x,y,          0b01111111);
        set_and_draw(x,y+1, 0b01000001);
        set_and_draw(x,y+2, 0b01000001);
        set_and_draw(x,y+3,          0b01000001);
        set_and_draw(x,y+4, 0b01111111);
    }

    void draw_P(char x, char y){

        set_and_draw(x,y,          0b01111111);
        set_and_draw(x,y+1,          0b00001001);
        set_and_draw(x, y+2, 0b00001001);
        set_and_draw(x,y+3, 0b00001001);
        set_and_draw(x,y+4, 0b00000110);
    }

    void draw_Q(char x, char y){

        set_and_draw(x,y,          0b00111111);
        set_and_draw(x,y+1, 0b01000001);
        set_and_draw(x,y+2, 0b01010001);
        set_and_draw(x,y+3,          0b00100001);
        set_and_draw(x,y+4,          0b01011110);
    }

    void draw_R(char x, char y){

        set_and_draw(x,y,          0b01111111);
        set_and_draw(x,y+1, 0b00001001);

```

```
    set_and_draw(x,y+2, 0b00011001);
    set_and_draw(x,y+3, 0b00101001);
    set_and_draw(x,y+4, 0b01000110);
}
```

```
void draw_S(char x, char y){
    set_and_draw(x,y, 0b01000110);
    set_and_draw(x,y+1, 0b01001001);
    set_and_draw(x,y+2, 0b01001001);
    set_and_draw(x,y+3, 0b01001001);
    set_and_draw(x,y+4, 0b00110001);
}
```

```
void draw_T(char x, char y){
    set_and_draw(x,y, 0b00000001);
    set_and_draw(x,y+1, 0b00000001);
    set_and_draw(x,y+2, 0b01111111);
    set_and_draw(x,y+3, 0b00000001);
    set_and_draw(x,y+4, 0b00000001);
}
```

```
void draw_U(char x, char y){
    set_and_draw(x,y, 0b00111111);
    set_and_draw(x,y+1, 0b01000000);
    set_and_draw(x,y+2, 0b01000000);
    set_and_draw(x,y+3, 0b01000000);
    set_and_draw(x,y+4, 0b00111111);
}
```

```
void draw_V(char x, char y){
    set_and_draw(x,y, 0b00001111);
    set_and_draw(x,y+1, 0b00110000);
    set_and_draw(x,y+2, 0b01000000);
    set_and_draw(x,y+3, 0b00110000);
    set_and_draw(x,y+4, 0b00001111);
}
```

```
void draw_W(char x, char y){
    set_and_draw(x,y, 0b00111111);
    set_and_draw(x,y+1, 0b01000000);
    set_and_draw(x,y+2, 0b00110000);
    set_and_draw(x,y+3, 0b01000000);
    set_and_draw(x,y+4, 0b00111111);
}
```

```
void draw_X(char x, char y){
    set_and_draw(x,y, 0b01100011);
    set_and_draw(x,y+1, 0b00010100);
    set_and_draw(x,y+2, 0b00001000);
}
```

```

        set_and_draw(x,y+3,      0b00010100);
        set_and_draw(x,y+4, 0b01100011);
    }

void draw_Y(char x, char y){

    set_and_draw(x,y,          0b00000011);
    set_and_draw(x,y+1,       0b00000100);
    set_and_draw(x, y+2, 0b01111000);
    set_and_draw(x,y+3, 0b00000100);
    set_and_draw(x,y+4, 0b00000011);

}

void draw_Z(char x, char y){

    set_and_draw(x,y,          0b01100001);
    set_and_draw(x,y+1, 0b01010001);
    set_and_draw(x,y+2, 0b01001001);
    set_and_draw(x,y+3,      0b01000101);
    set_and_draw(x,y+4, 0b01000011);

}

/* NUMBERS */

void draw_0(char x, char y){

    set_and_draw(x,y,          0b01111111);
    set_and_draw(x,y+1, 0b01010001);
    set_and_draw(x,y+2, 0b01001001);
    set_and_draw(x,y+3,      0b01000101);
    set_and_draw(x,y+4, 0b01111111);

}

void draw_1(char x, char y){

    set_and_draw(x,y, 0b00000000);
    set_and_draw(x,y+1,      0b01000010);
    set_and_draw(x,y+2, 0b01111111);
    set_and_draw(x,y+3, 0b01000000);
    set_and_draw(x,y+4, 0b00000000);

}

void draw_2(char x, char y){

    set_and_draw(x,y,          0b01000010);
    set_and_draw(x,y+1, 0b01100001);
    set_and_draw(x,y+2, 0b01010001);
    set_and_draw(x,y+3,      0b01001001);
    set_and_draw(x,y+4, 0b01000110);

}

void draw_3(char x, char y){

    set_and_draw(x,y,          0b00100001);
    set_and_draw(x,y+1, 0b01000001);

```



```

        set_and_draw(x,y+2, 0b01000101);
        set_and_draw(x,y+3, 0b01001011);
        set_and_draw(x,y+4, 0b00110001);
    }
    void draw_4(char x, char y){

        set_and_draw(x,y, 0b00011000);
        set_and_draw(x,y+1, 0b00010100);
        set_and_draw(x,y+2, 0b00010010);
        set_and_draw(x,y+3, 0b01111111);
        set_and_draw(x,y+4, 0b00010000);

    }
    void draw_5(char x, char y){

        set_and_draw(x,y, 0b00100111);
        set_and_draw(x,y+1, 0b01000101);
        set_and_draw(x,y+2, 0b01000101);
        set_and_draw(x,y+3, 0b01000101);
        set_and_draw(x,y+4, 0b00111001);

    }
    void draw_6(char x, char y){

        set_and_draw(x,y, 0b00111100);
        set_and_draw(x,y+1, 0b01001010);
        set_and_draw(x,y+2, 0b01001001);
        set_and_draw(x,y+3, 0b01001001);
        set_and_draw(x,y+4, 0b00110000);

    }
    void draw_7(char x, char y){

        set_and_draw(x,y, 0b00000001);
        set_and_draw(x,y+1, 0b01110001);
        set_and_draw(x,y+2, 0b00001001);
        set_and_draw(x,y+3, 0b00000101);
        set_and_draw(x,y+4, 0b00000011);

    }
    void draw_8(char x, char y){

        set_and_draw(x,y, 0b00110110);
        set_and_draw(x,y+1, 0b01001001);
        set_and_draw(x,y+2, 0b01001001);
        set_and_draw(x,y+3, 0b01001001);
        set_and_draw(x,y+4, 0b00110110);

    }
    void draw_9(char x, char y){

        set_and_draw(x,y, 0b00000110);
        set_and_draw(x,y+1, 0b01001001);
        set_and_draw(x,y+2, 0b01001001);
        set_and_draw(x,y+3, 0b00101001);
        set_and_draw(x,y+4, 0b00011110);
    }

```

}