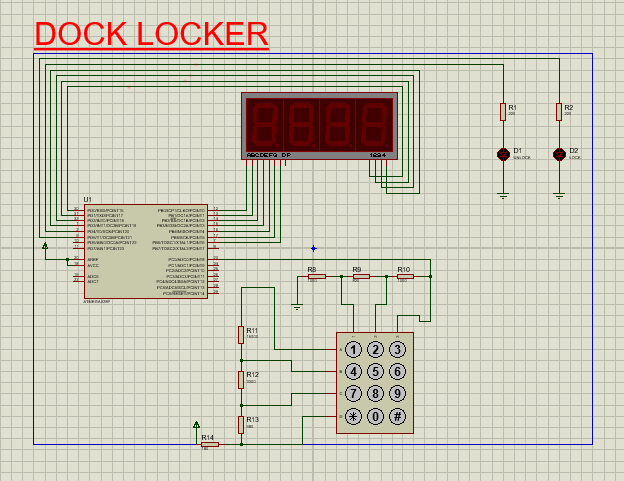
# วงจรประตูนิรภัย

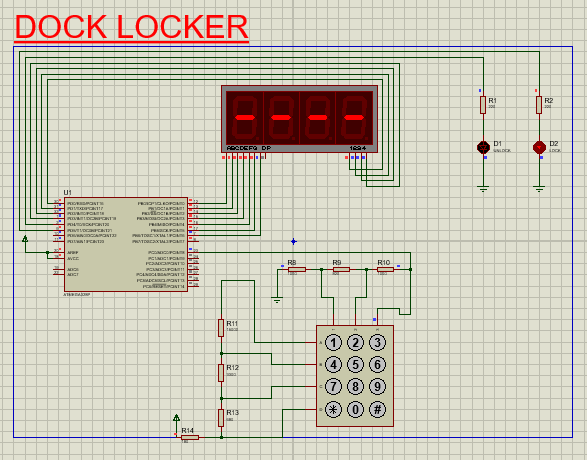
## สมาชิกในกลุ่ม

* นาย
* นางสาว

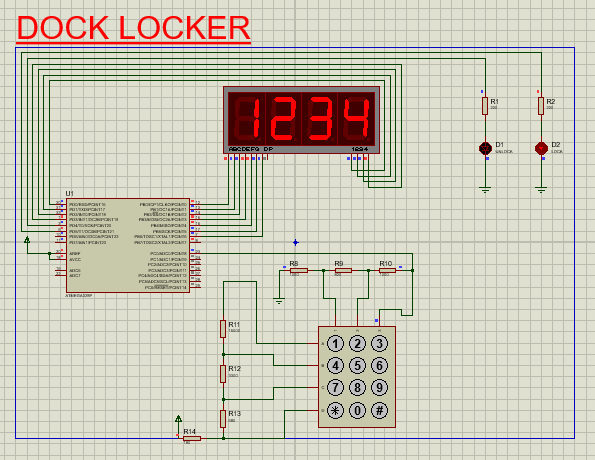
## โครงสร้างระบบ



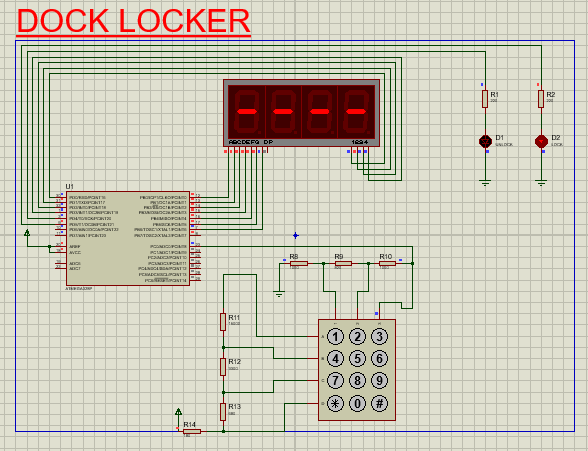
## หลักการทำงาน

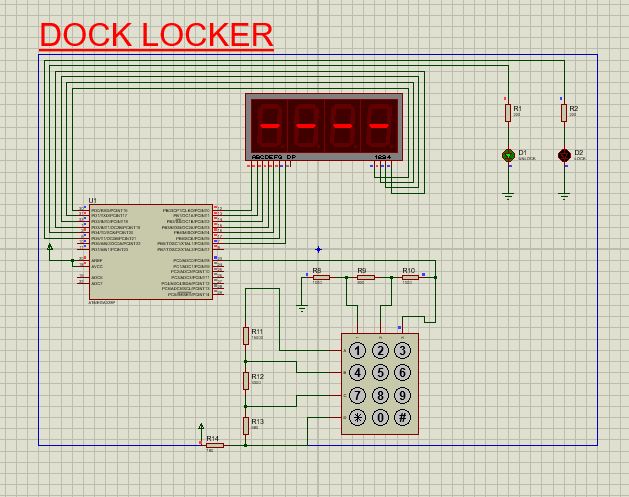


เมื่อเปิดโปรแกรม

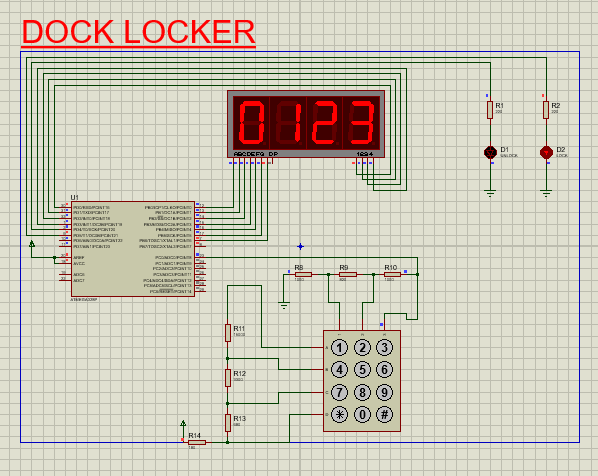


หลักจากที่กด password





หลังจากกด ตรวจสอบ password ( ปุ่ม # ) หาก password ตรงออกเขียว หากไม่ ออกแดง



เมื่อกดตั้งค่า password ( ปุ่ม \* ) หลังจากที่กดตั้งค่าครบ 4 หลักโปรแกรมจะกลับไปหาเริ่มต้นโปรแกรม

## สรุปเทคนิคที่ใช้

## Source code

#include <inttypes.h>

#include <avr/io.h>

#include <avr/interrupt.h>

#include <avr/sleep.h>

#define TURN\_ALL\_LED\_OFF 0xFF

#define INTERVAL\_time 65528

unsigned int ACCUMULATED, DIVISOR, QUOTIENT, DECODE, DECODED, STATE, SWITCH;

volatile unsigned int DISPLAY\_COL, INPUT\_COL, SET\_PWD\_COL, DISPLAY\_PWD[4] = {16,16,16,16}, SET\_PWD[4] = {0,1,2,3}, SET\_TIMEOUT\_PWD;

unsigned char TABLE7SEG[] ={

0b00111111, // 0

0b00000110, // 1

0b01011011, // 2

0b01001111, // 3

0b01100110, // 4

0b01101101, // 5

0b01111101, // 6

0b00100111, // 7

0b01111111, // 8

0b01101111, // 9

0b01110111, // A

0b11111100, // b

0b00111001, // C

0b00111111, // D

0b01111001, // E

0b01110001, // F

0b01000000, // -

0b00000000 }; // turn off all segment

ISR(TIMER1\_OVF\_vect){

TCNT1 = INTERVAL\_time;

SET\_TIMEOUT\_PWD++;

if(SWITCH){

PORTD = 1 << DISPLAY\_COL;

PORTB = ~TABLE7SEG[SET\_PWD[DISPLAY\_COL]];

DISPLAY\_COL = (DISPLAY\_COL + 1)%4;

}

else{

PORTD = 1 << DISPLAY\_COL;

PORTB = ~TABLE7SEG[DISPLAY\_PWD[DISPLAY\_COL]];

DISPLAY\_COL = (DISPLAY\_COL + 1)%4;

}

if(STATE) PORTD |= 0b00010000; // to check password ture or false

else PORTD |= 0b00100000;

if(SET\_TIMEOUT\_PWD > 1000){ // when over at 1000 time will set STATE = 0 and SET\_TIMEOUT\_PWD for back to normal state

STATE = 0;

SET\_TIMEOUT\_PWD = 0;

}

}

ISR(ADC\_vect){

if(ADCH > 9){

ACCUMULATED += ADCH;

DIVISOR++;

}

else {

ACCUMULATED = 0;

DIVISOR = 0;

}

if(DIVISOR == 10){

QUOTIENT = ACCUMULATED/DIVISOR;

if ((QUOTIENT>9) && (QUOTIENT<16)) DECODED = 1 ;

else if ((QUOTIENT>18) && (QUOTIENT<26)) DECODED = 2 ;

else if ((QUOTIENT>29) && (QUOTIENT<37)) DECODED = 3 ;

else if ((QUOTIENT>46) && (QUOTIENT<53)) DECODED = 4 ;

else if ((QUOTIENT>74) && (QUOTIENT<81)) DECODED = 5 ;

else if ((QUOTIENT>99) && (QUOTIENT<107)) DECODED = 6 ;

else if ((QUOTIENT>133) && (QUOTIENT<141)) DECODED = 7 ;

else if ((QUOTIENT>169) && (QUOTIENT<177)) DECODED = 8 ;

else if ((QUOTIENT>192) && (QUOTIENT<199)) DECODED = 9 ;

else if ((QUOTIENT>212) && (QUOTIENT<220)) DECODED = 10 ;

else if ((QUOTIENT>228) && (QUOTIENT<236)) DECODED = 0 ;

else if ((QUOTIENT>236) && (QUOTIENT<244)) DECODED = 11 ;

else DECODED = 16 ;

if(DECODED < 10){ // hit keypad

if(SWITCH){

SET\_PWD[SET\_PWD\_COL] = DECODED;

SET\_PWD\_COL = SET\_PWD\_COL + 1;

if(SET\_PWD\_COL == 4){

SET\_PWD\_COL = 0;

SWITCH = 0;

}

}

else{

DISPLAY\_PWD[INPUT\_COL] = DECODED;

INPUT\_COL = (INPUT\_COL + 1)%4;

}

}

if(DECODED == 10){ // set password

SWITCH = 1;

}

if(DECODED == 11){ // check password

if(DISPLAY\_PWD[0] == SET\_PWD[0] &&

DISPLAY\_PWD[1] == SET\_PWD[1] &&

DISPLAY\_PWD[2] == SET\_PWD[2] &&

DISPLAY\_PWD[3] == SET\_PWD[3]){

STATE = 1;

DISPLAY\_PWD[0] = 16;

DISPLAY\_PWD[1] = 16;

DISPLAY\_PWD[2] = 16;

DISPLAY\_PWD[3] = 16;

}

else{

STATE = 0;

DISPLAY\_PWD[0] = 16;

DISPLAY\_PWD[1] = 16;

DISPLAY\_PWD[2] = 16;

DISPLAY\_PWD[3] = 16;

}

}

}

}

int main(void)

{

DDRB = 0xFF;

DDRD = 0xFF;

PORTB = TURN\_ALL\_LED\_OFF;

ADMUX = 0b00100000;

//[7:6] AREF Pin

//[5] ADLR1

//[3:0] ADC0

ADCSRA = 0b10101101;

ADCSRB = 0b00000110; // Time1 OVF

TIMSK1 = 0x01;

TCCR1A = 0x00;

TCCR1B = 0x05;

cli();

TCNT1 = INTERVAL\_time;

sei();

DIVISOR = 0;

DISPLAY\_COL = 0;

INPUT\_COL = 0;

STATE = 0;

SWITCH = 0;

while(1);

}