**Cng 336**

**Lab 5 Prework**

**Umut Yıldırım - 2315612**

**Ulaş Cem Erten – 2256006**

**5.2.2**

Vref and number of bits because step size = Vref/(2^n) where n is number of bits.

**5.2.3**

1) Sample-and-hold circuit samples and stores the analog input.

2) The logic starts with generating a digital value to D/A.

3) At the comparator, Vout is compared with Vin.

4) When Vin > Vout, Cout=‘1’, which causes the logic to freeze Most Significant Bit (MSB) at ‘1’, and generate a halfway (midrange) signal between Vcc/2 and Vcc.

5) If vin ≤ vout, Cout=‘0’, which causes the logic to freeze MSB at ‘0’, and generate a halfway (midrange) signal between 0 and Vcc/2 e.g. 010000 for a 6-bit A/D

6)Same processes from second step to fifth step repeated for 2nd MSB and then for 3rd MSB until LSB.

7) When approximation sequence is complete, the A/D control enables the digital signal to the digital output port.

**5.2.4**

Conversion time = clock cycles \* number of bits \* period

Conversion time = 4 \* 10 \* 1/(800\*10^6) = 50 microseconds

**5.2.5**

**Assembly Code**

.include "m128def.inc"

.equ ones = 0xFF

.equ zeros = 0

.org 0x00

.cseg

.macro initstack

ldi r16,low(ramend)

out spl,r16

ldi r16,high(ramend)

out sph,r16

.endmacro

initstack

main:

ldi r16,zeros

sts ddrf,r16 ;portf is input for ADC

ldi r16,ones

out ddra,r16;

out ddrb,r16;A an B ports are now output

ldi r16, 0xA7;converter set up

out ADCSRA,r16;enabling adc in free running mode with clock 128

ldi r16, 0x00

out ADMUX,r16;Vref set externally

a\_d\_c:

sbi ADCSRA, ADSC;start conversion

cont:;polling here untill ADIF bit is set

sbis ADCSRA, ADIF;If ADIF set then conversion completed

rjmp cont

sbi ADCSRA, ADIF;Clear ADIF flag

in r16,ADCL;load our value

out porta,r16;send to output port

in r16,ADCH;load higher bit

out portb,r16;send to output port

rjmp a\_d\_c;do it again

**MicroC code:**

int main (void)

{

DDRA = 0xFF;

DDRB = 0xFF;

DDRF = 0;

ADCSRA = 0xA7;

ADMUX = 0x00;

while (1){

ADCSRA |= (1<<ADSC);

while((ADCSRA&(1<<ADIF))==0);

PORTA = ADCL;

PORTB = ADCH;

}

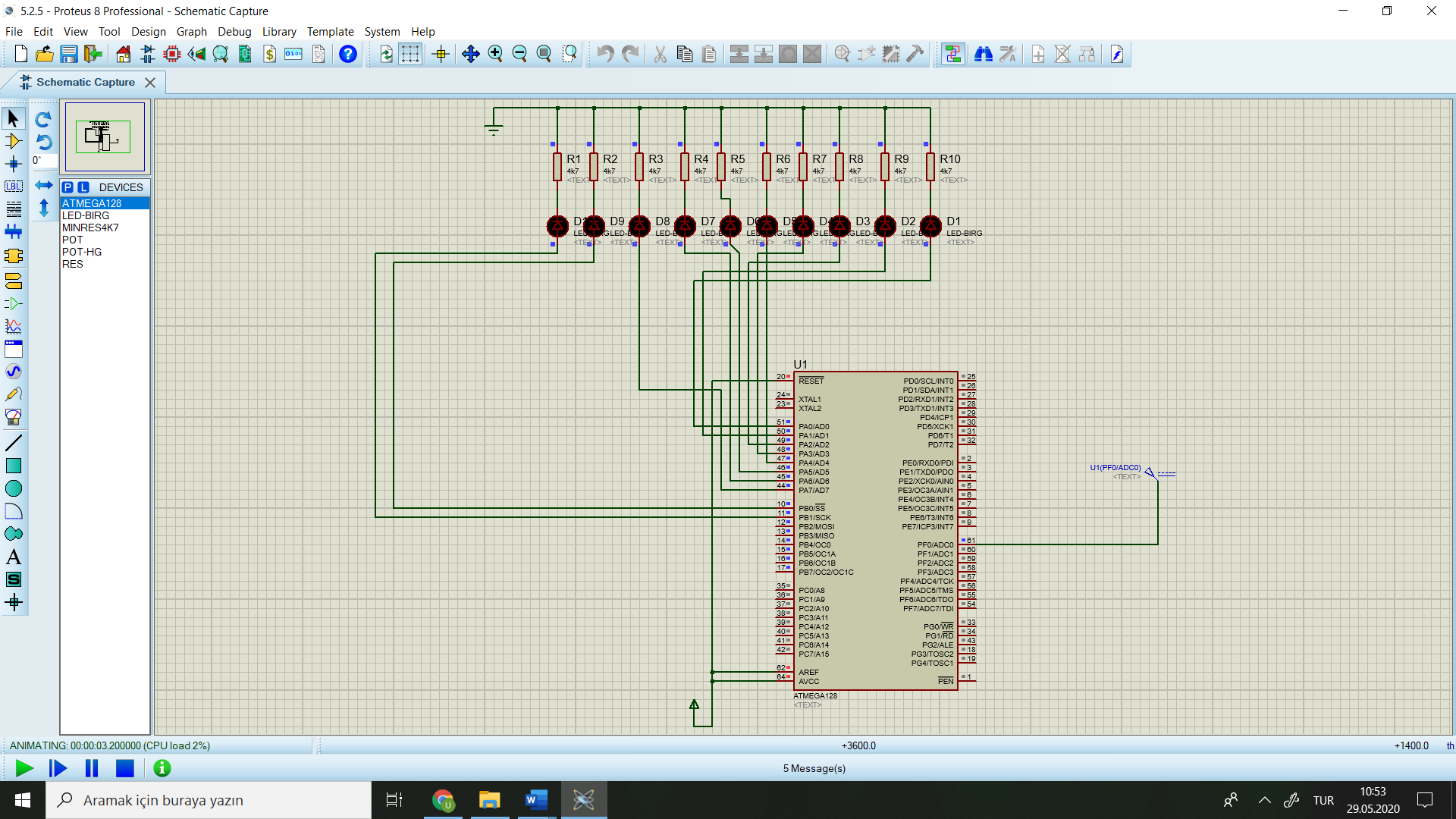
return 0;

}

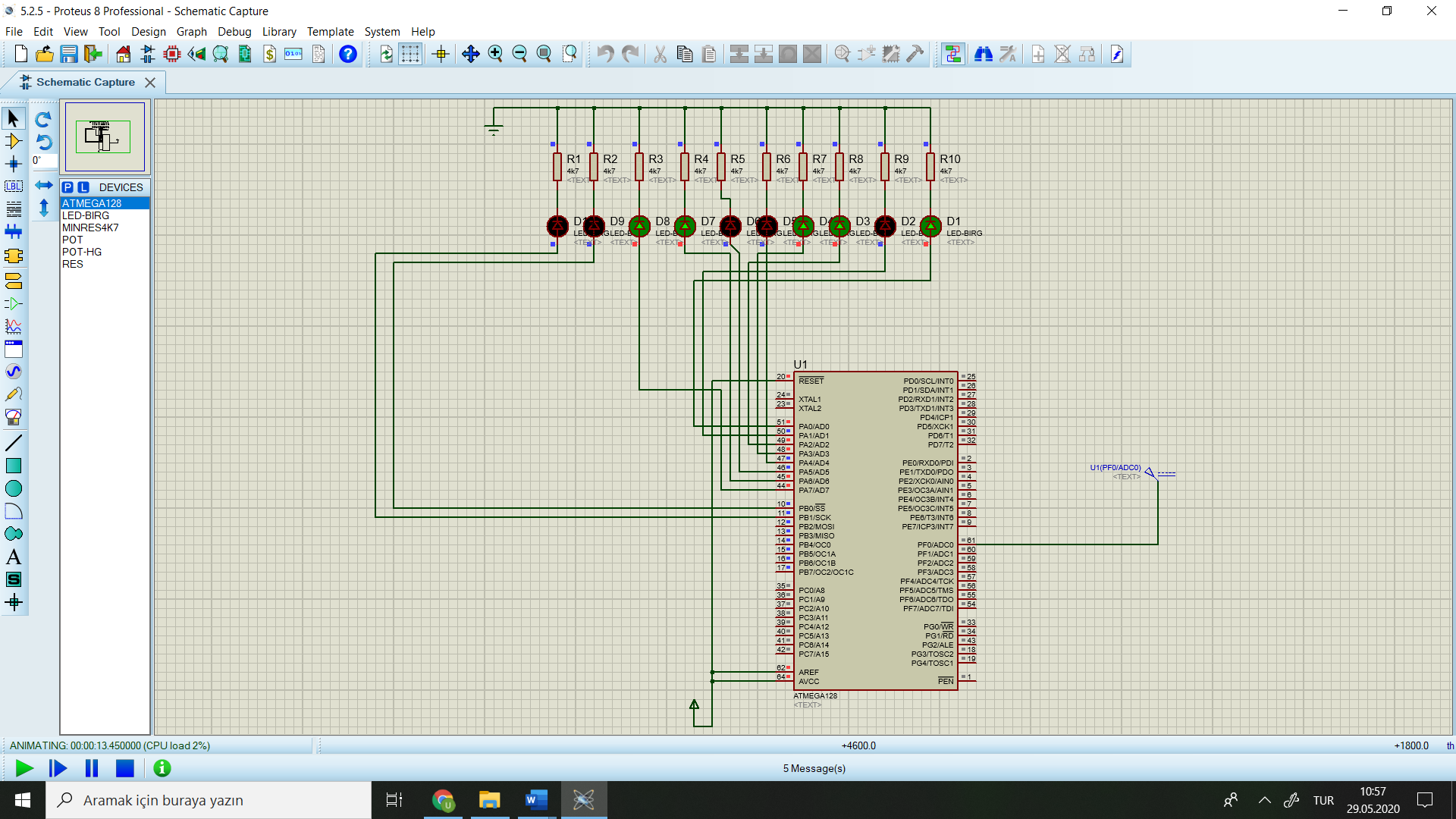
**Example screen shots :**

Step size = 5V / 2^10 = 4.88 mV

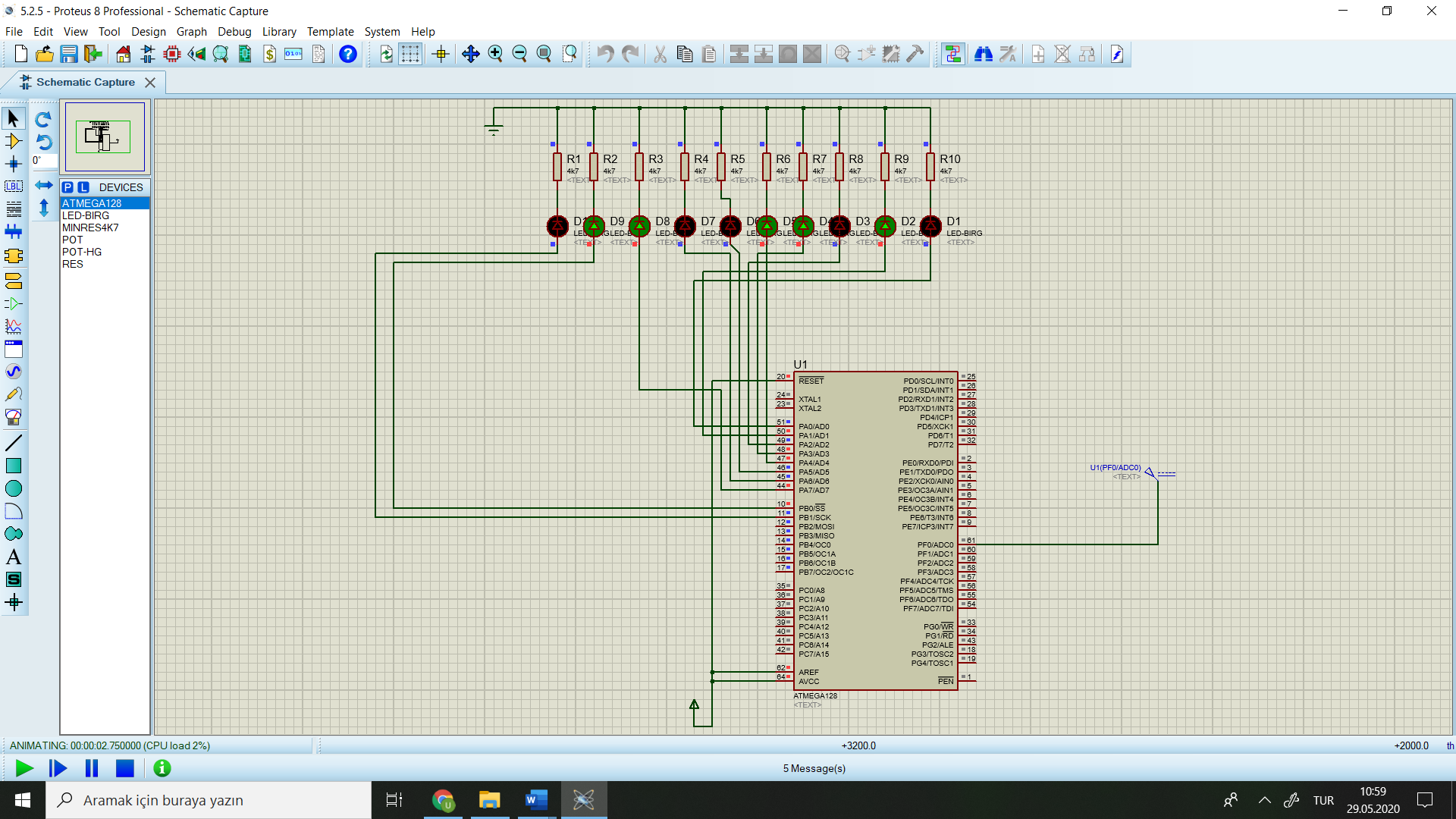
When 0 V - > 0V/4.88 mV = 0b0000000000



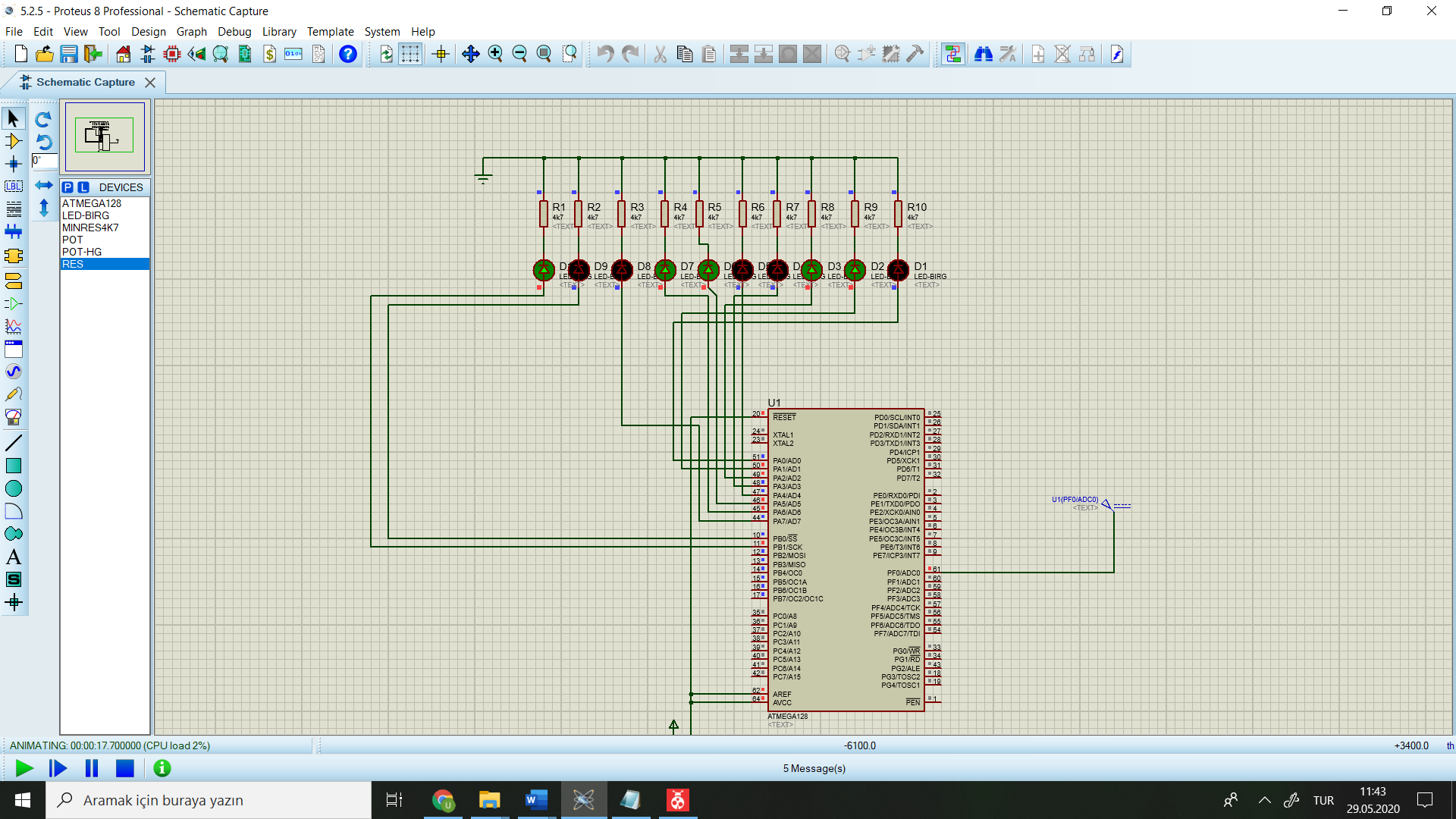
When 1V - > 1V / 4.88 = 0b0011001101



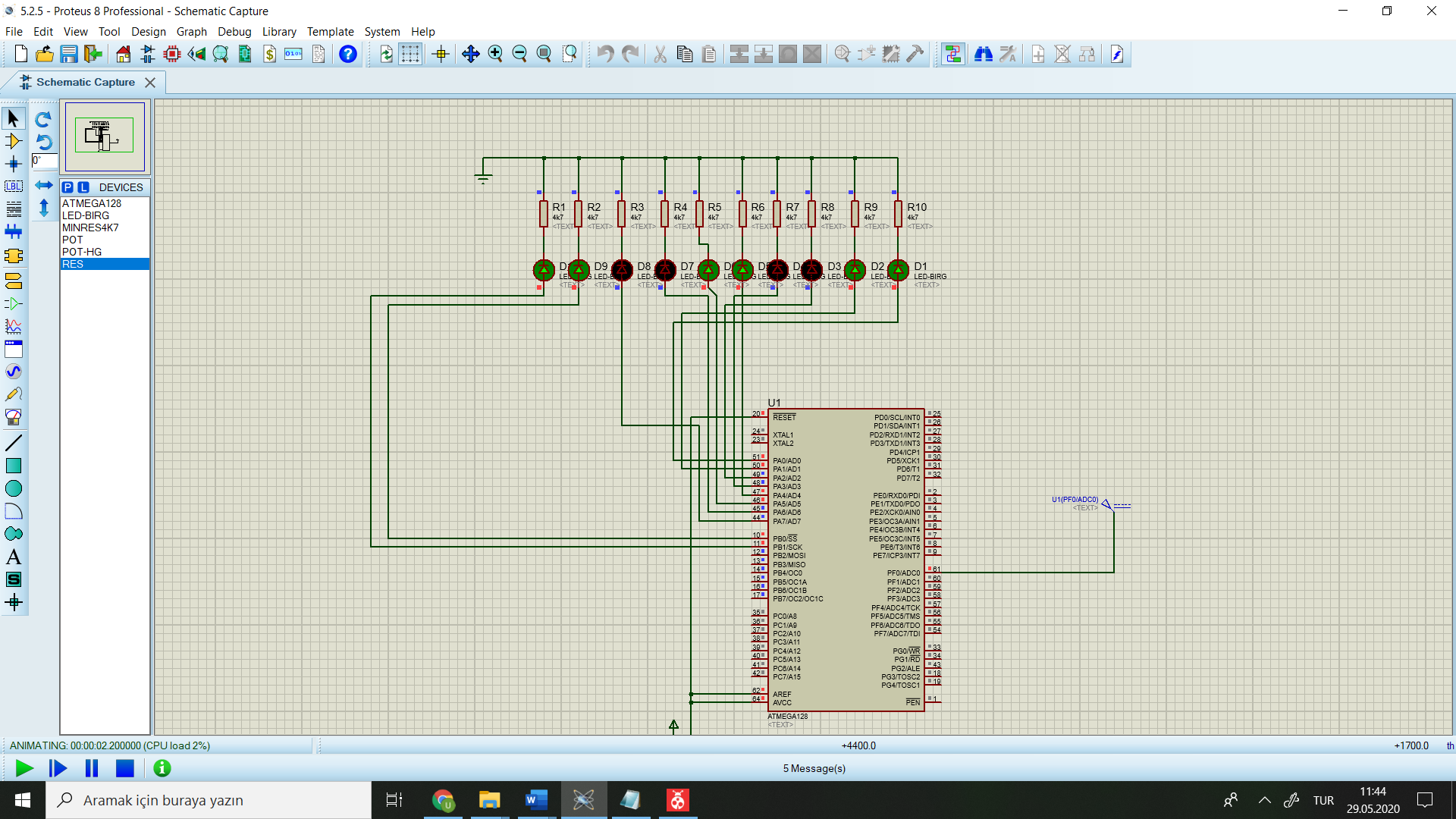
When 2V - > 2V/4.88mV = 0b0110011010



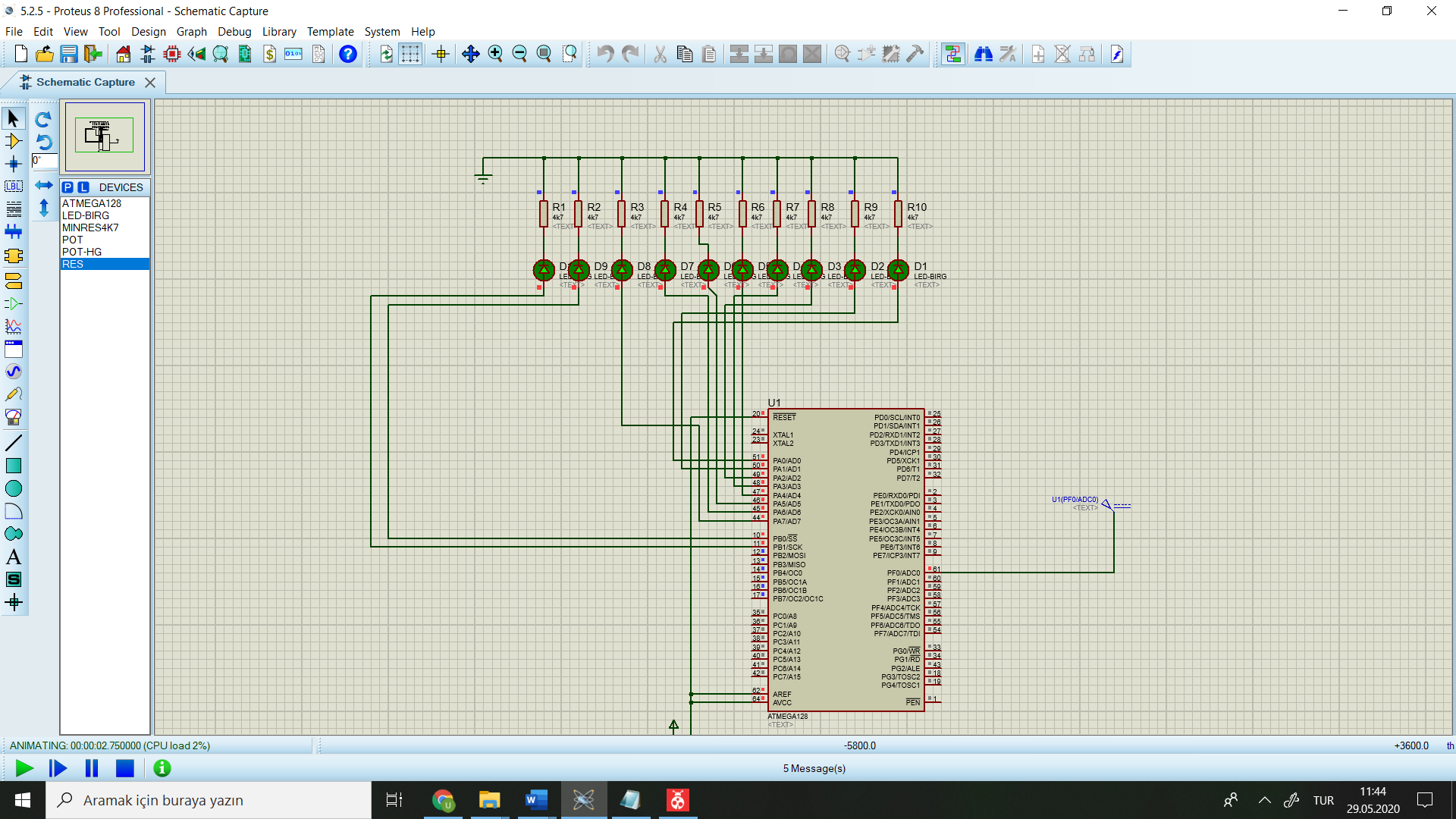
When 3V - > 3V/4.88mV = 0b1001100110



When 4V - > 4V/4.88mV = 0b1100110011



When 5V - > 5V/4.88mV = 0b1111111111



**5.2.6**

a) when input is 00000000 -> V0 Span = 5\*0 = 0

when input is 11111111 -> V0 Span = 5 \* (1/2+1/4+1/8+1/16+1/32+1/64+1/128+1/256) = 4.98

b) Vo Resolution = Span / 2^n = 4.98/2^8 = 19.45 mV

c) A [1:8] = 10000000 -> V0 = Vref/2 = 4.98 / 2 =2.49 V

**5.2.7**

**Assembly Code**

.include "m128def.inc"

.org 0x00

ldi r16, 0xFF

out DDRB, r16

continue:inc r16

out PORTB, r16

rjmp continue

**Screen Shots**

