7 Segment

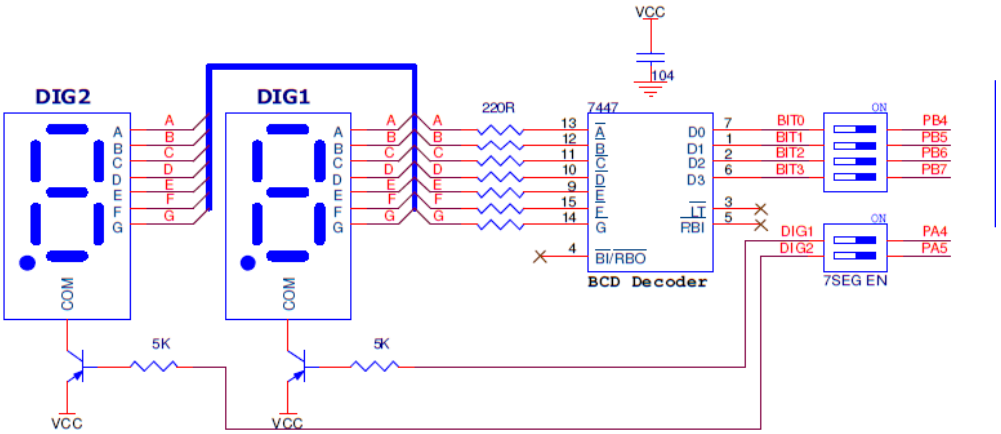
Nada Elshawy & Noha Elmenbawy

الفكره:

تشغيل عدد 2 من ال Seven Segment

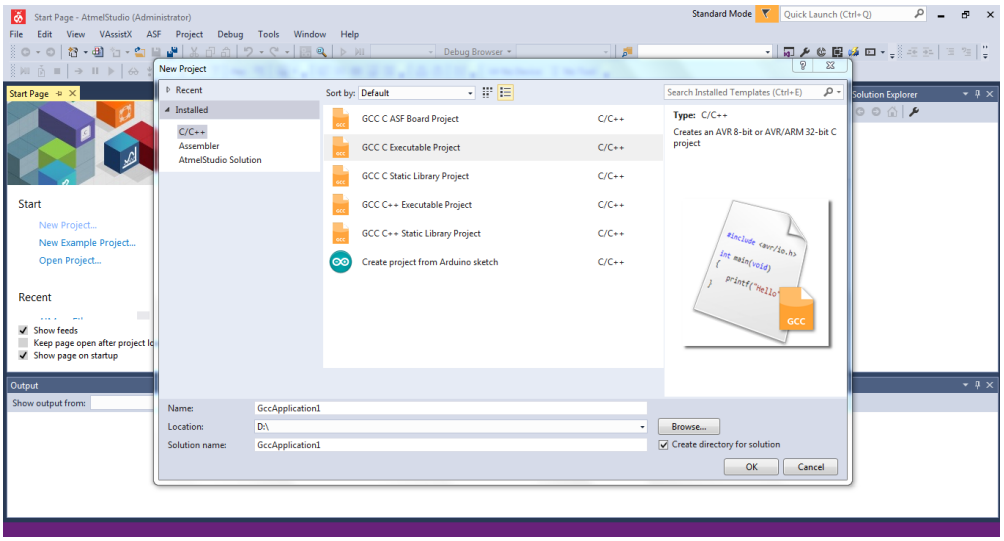
المخطط (schematic):

يتم استخدامDecoder +2 من المقاطعه السباعيه (7Segment)

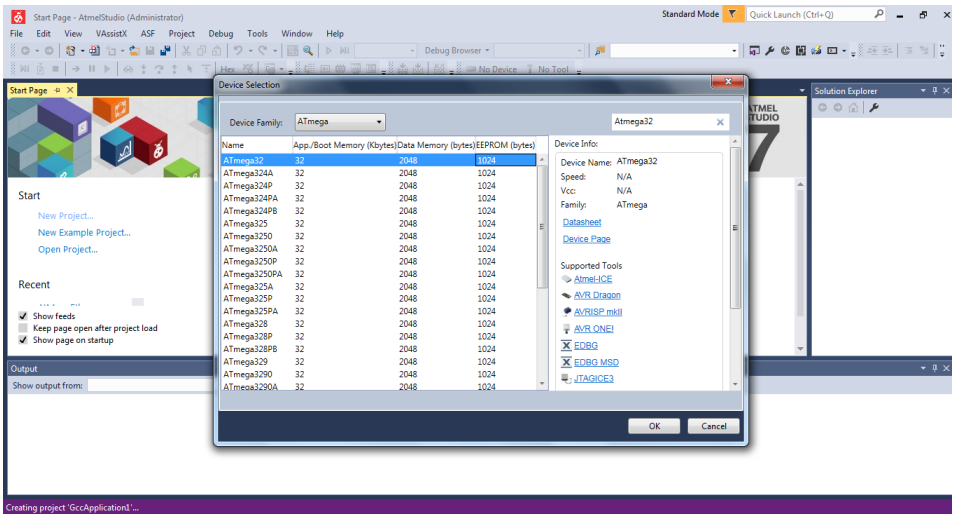


البرنامج المستخدم لكتابه الكود:

Atmel Studio 7.0



يتم انشاء مشروع جديد وكتابه اسم له وتحديد المكان الذى سوف يتم حفظ المشروع به ويتم اختيار GCC C Executable project



يتم اختيار اسم الميكروكنترولر المستخدم لذا سوف نختار Atmega32

الادوات المستخدمه (Hardware) :

- Avr development Kit

http://ram-e-shop.com/oscmax/catalog/product_info.php?products_id=2854

الكود :

1

```
/*
 * seven_seg.c
 *
 * Created: 2/6/2017 4:57:19 PM
 *  Author: NadaHassan
 */

#define F_CPU 8000000ul
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>              // include the interrupt header file

void print_num(uint8_t num);            // prototype of printing number in seven segment
void init_sevenSeg(void);               // prototype of initializing the seven segment
void init_timer(void);                  // prototype of initializing the timer

uint8_t x=0;                            // global variable of the number to be printed on seven segment

int main(void)
{
    init_sevenSeg();                    // initialize seven segment
    init_timer();                       // initialize timer1

    while(1)
    {
        print_num(x);                   // print x in seven seg
```

2

```
    }
}

void init_sevenSeg(void){
    DDRA |=(1<<PA5) |(1<<PA4);          // set pin4 and pin5 in port A as output
    DDRB |= 0xf0;                       // set the for most significant bits as output
}
void print_num(uint8_t num){            // function for printing numbers
    uint8_t left=0;                     // variable for tens digits
    uint8_t right=0;                    // variable for ones digits
    left = num/10;                      // to get the tens
    right = num %10;                    // to get the ones

    PORTA |=(1<<PA4);                   // turn on the left seven seg on the kit
    PORTA &=~(1<<PA5);                  // turn off the right seven seg on the kit
    PORTB =(left<<4);                   // printing tens digit

    _delay_ms(10);                      // wait 10 ms
    PORTA &=~(1<<PA4);                  // turn off the left seven seg on the kit
    PORTA |=(1<<PA5);                   // turn off the right seven seg on the kit
    PORTB =(right<<4);

    _delay_ms(10);                      // wait
```

3

```
}
void init_timer(void){                  // initialize timer function
    TIMSK |=(1<<TOIE1);                 // enable interrupt flage
    TCCR1B |=(1<<CS11);                 // using prescaler /8
    sei();                              // enable global interrupt
}
ISR(TIMER1_OVF_vect){                   // interrupt service routine
    if (x==99)                          // in case the number reaches 99
    {
        x=0;                            // reset the value of x to 0 because we only have two seven se on the kit
    }else{
        x++;                            // increment x value
    }
}
```

```
#define F_CPU 8000000UL
```

هذا الامر يحدد سرعه المعالج الداخليه (التردد الذى يعمل به المعالج داخل الميكروكنترولر)

فى بدايه داله الmainيتم استدعاء الداله التى تقوم بعمل Initialization للمقاطعه السباعيه حيث يتم تحديد الاطراف المسؤله عن تشغيل المقاطعه السباعيه وتحديدها كخرج كما يتم ضبط الاطراف المتصله بال Encoder كخرج ايضا

ثم بعد ذلك يتم استدعاء الداله الخاصه بتفعيل وضبط التايمر ويتم تنشيط الانتربت بداخلها ايضا ..فعند وصول التايمر ل 256 يحدث انتربت ويتم استدعاء الداله ISRلتنفيذ مابها من اوامر

داخل ال while(1) يتم استدعاء الداله المسؤله عن اظهار الارقام على ال 7Segment حيث ان الاحاد يظهر على المقاطعه السباعيه اليمنى والعشرات على اليسرى

المراجع المستخدمه :

- The AVR microcontroller and embedded system using assembly and c
- Simply Avr

Links:

- https://docs.google.com/file/d/0B5_mAdKvdKTlQlBPS2pwbE9ONFE/view
- http://www.mediafire.com/file/mcd4k2r67j1r1nm/Simply_AVR.zip