

# keyPadAVR

Övning: Knappsats på Atmega328p

Kurs

Programmering av inbyggda system

**Johan Kämpe**

UIS16

2017-04-11

## Innehåll

1	Inledning.....	3
1.1	Syfte.....	3
1.2	Bakgrund .....	3
1.2.1	Knappsats .....	3
1.3	Länkar .....	3
2	Genomförande och resultat .....	4
2.1	Använd programvara och hårdvara.....	4
2.2	Avgränsningar.....	4
2.3	Metod .....	5
2.3.1	Skrivning av källkod, kompilering och uppladdning till MCU .....	5
2.3.2	Fastställande av relation mellan knappar och anslutningar på knappsatsen .....	5
2.3.3	Indikation av knapptryck på knappsats .....	5
2.4	Anslutning av hårdvara.....	6
2.5	Projektets kod .....	7
2.5.1	Macron .....	7
2.5.2	Variabler .....	8
2.5.3	Inkluderade standardbibliotek .....	9
2.5.4	Funktioner .....	9
2.5.5	Main-funktionen.....	10
3	Bilagor.....	11
3.1	Programmets källkod, main-funktion.....	11
3.2	Programmets källkod, funktioner.....	12

# 1 Inledning

## 1.1 Syfte

Syftet med övningen är få en numerisk knappsats att fungera på en Atmega328p mikroprocessor (MCU). När en siffra trycks in på knappsatsen ska en lysdiod blinka lika många gånger som värdet på siffran.

Övningen som beskrivs i rapporten kan också komma att kallas *projektet* i rapporten.

## 1.2 Bakgrund

### 1.2.1 Knappsats

Knappsatsen som används i projektet består av fyra rader och tre kolumner av knappar enligt följande tabell:

1	2	3
4	5	6
7	8	9
*	0	#

Tecknen i tabellen representerar en knapp med samma värde på knappsatsen.

På knappsatsen sitter sju stycken anslutningar. När en knapp trycks ned sluts kretsen mellan två av dessa anslutningar. På detta sett kan det avgöras vilken av knapparna som har tryckts ned.

## 1.3 Länkar

**GitHub-sida för projektet**

<https://github.com/GoblinDynamiteer/progInbyggdaSystem/tree/master/Projekt/keypadAVR>

**Demovideo, YouTube**

<https://www.youtube.com/watch?v=5uytAdWNXpI>

## 2 Genomförande och resultat

### 2.1 Använd programvara och hårdvara

#### Programvaror

- Atmel Studio 7.0
- Microsoft Excel
- AVRDUDE, AVR Downloader/UploaDEr
- AVRDUDESS 2.4 – GUI för AVRDUDE

#### Hårdvara

- Atmel Atmega328p MCU. Hänvisas till som mikroprocessor eller MCU i rapporten.
- 3x4 knappars knappsats. Siffror 0 - 9, samt # och \*
- Resistor 10 k $\Omega$ , 3 st. För pull-down av knappar.
- Kablar för anslutning av MCU-pins till knappsats
- Sparkfun PGM-09825 Pocket AVR Programmer
- Multimeter
- Blå LED
- Resistor 330  $\Omega$ , för anslutning av LED
- Breadboard, anslutningsplatta

### 2.2 Avgränsningar

Endast siffror 1-9 på knappsatsen används.

Således används *inte* knappar 0, #, \* i projektet. Dessa ligger på den fjärde raden på knappsatsen.

## 2.3 Metod

### 2.3.1 Skrivning av källkod, kompilering och uppladdning till MCU

Programmets källkod skrevs uteslutande i utvecklingsmiljön Atmel Studio. Källkoden är skriven i programspråket C.

Vid kompilering av koden skapas en fil med filändelsen *hex*. Denna fil laddas upp till mikroprocessorn (MCU, MicroController Unit) med datorprogrammet AVRDUDE.

### 2.3.2 Fastställande av relation mellan knappar och anslutningar på knappsatsen

En multimeter användes för att mäta kontakt mellan två de sju anslutningarna på knappsatsen. När en knapp trycktes ned, och multimetern indikerade kontakt, noterades de två anslutningarna i en tabell i datorprogrammet Excel.

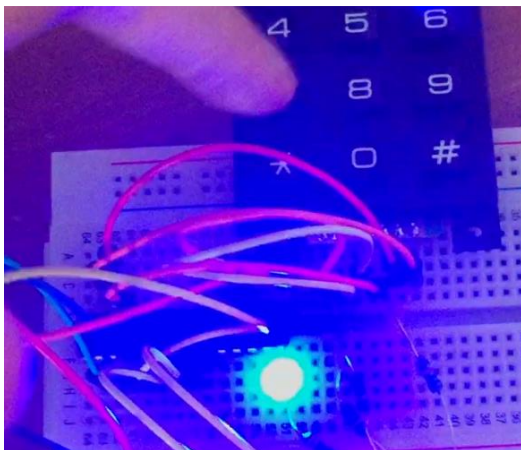
Resultatet blev följande:

Anslutning	3	1	5
2	1	2	3
7	4	5	6
6	7	8	9
4	*	0	#

Detta visar att när till exempel knapp 3 trycks ned på knappsatsen, så sluts kretsen mellan anslutning 5 och 2.

### 2.3.3 Indikation av knapptryck på knappsats

För att indikera vilken siffra som har tryckts på knappsatsen ansluts en blå LED (Light Emitting Diode, lysdiod) till mikroprocessorn. Lysdioden blinkar lika många gånger som värdet på siffran.



Figur 1 LED-indikator för knapptryck

## 2.4 Anslutning av hårdvara

Uppkoppling av komponenter gjordes på en kopplingsplatta, en så kallad *breadboard*. Kablar användes för att ansluta knappsatsen till mikroprocessorn.

Programmeringsenheten, *Pocket AVR Programmer*, användes som strömkälla. Spänningsmatning 5V anslöts till VCC (pin 29) och jord till GND (pin 22).

Lysdioden som används för att indikera vilken knapp som har tryckts på, på knappsatsen, kopplades till mikroprocessorns anslutning PC5 (pin 28).

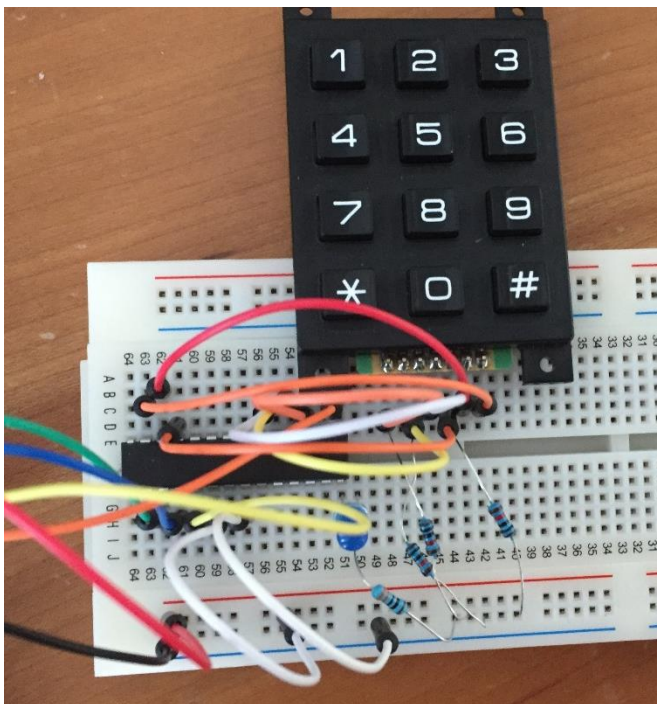
Lysdiodens ben anslöts till jord via en resistor med resistans 330Ω.

Knappsatsens anslutningar kopplades till mikroprocessorn enligt följande:

Knappsats anslutning	Atmega328p Anslutning
1	PD1 (pin 3)
2	PD2 (pin 4)
3	PD3 (pin 5)
5	PD5 (pin 11)
6	PD6 (pin 12)
7	PD7 (pin 13)

Anslutning 4 på knappsatsen användes inte, då denna är för den nedersta raden, som valdes att inte användas i projektet.

Till varje anslutning för knappsatsens kolumner (anslutningar 3, 1 och 5) kopplades en pull-down resistor med resistans 10kΩ.



Figur 2 Fotografi på ansluten knappsats till Atmega328p

## 2.5 Projektets kod

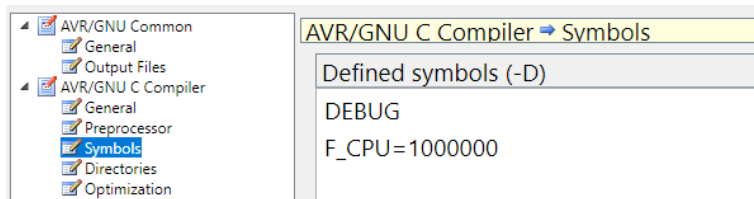
Källkoden kan ses i rapportens bilagor.

### 2.5.1 Macron

*Macron som är definierade i källkoden*

<b>#define ROWS 3</b>	Antal rader på knappsatsen, den fjärde raden används ej i projektet.
<b>#define COLS 3</b>	Antal kolumner på knappsatsen.
<b>enum { ROW1_PIN = (0b00000100), ROW2_PIN = (0b10000000), ROW3_PIN = (0b01000000) };</b>	Används för att sätta anslutningar till rader på knappsats till 1 / 0.
<b>enum { COL1_PIN = PD3, COL2_PIN = PD1, COL3_PIN = PD5 };</b>	Används för att testa om anslutningar till kolumner på knappsats är 1 / 0.

Macrot `F_CPU` behövs vara angivet för standardbiblioteket `delay.h`, detta sätts i kodprojektets egenskaper till `F_CPU=1000000`, i programmet Atmel Studio.



Figur 3 Kodprojektets egenskaper – Defined symbols

## 2.5.2 Variabler

*Variabler i källkoden, sätts i main-funktionen*

```
int columnPin[3] = {  
COL1_PIN,  
COL2_PIN,  
COL3_PIN  
};
```

Array för anslutningar på MCU som är anslutna till knappsatsens kolumner. Används i programmetts huvudloop för att testa knapptryckningar.

```
int rowPin[3] = {  
ROW1_PIN,  
ROW2_PIN,  
ROW3_PIN  
};
```

Array för anslutningar på MCU som är anslutna till knappsatsens rader. Används i programmetts huvudloop för att testa knapptryckningar.

```
int num = 1;
```

Används i programmetts huvudloop för att avgöra vilken knapp som tryckts på, på knappsatsen.



### 2.5.3 Inkluderade standardbibliotek

#### **<avr/io.h>**

Diverse I/O definitioner för den aktuella mikroprocessorn som används i kodprojektet.

#### **<util/delay.h>**

För funktionen `_delay_ms()` som pauserar programmet i angiver antal millisekunder.

### 2.5.4 Funktioner

*Egenskrivna funktioner i källkoden*

#### **void blinkLed(int n);**

Blinkar lysdioden som är kopplad till mikroprocessorns anslutning PC5. Lysdioden blinkar *n* antal gånger, vilket skickas som argument vid anrop av funktionen.

#### **int checkPin(int pin);**

Returnerar 1 eller 0, beroende på om anslutning *pin* på mikroprocessorns PD-anslutningar är hög eller låg. Argumentet *pin* skickas som argument vid anrop av funktionen.

### 2.5.5 Main-funktionen

*I detta kapitel beskrivs grundläggande vad som händer i programmets kod*

Initialt sätts mikroprocessorns anslutning PC5 till att användas som utgång:

```
DDRC = 0b00100000;
```

DDR innebär *Data Direction Register* och C den samling av åtta anslutningar som tillhör PC0 – PC7 på mikroprocessorn.

Alla åtta anslutningar sätts samtidigt, en etta innebär utgång, och en nolla ingång. 0b001000000 innebär att anslutning 5 sätts till utgång och alla andra som ingångar.

PC5 används för att styra lysdioden som används för indikering av vilken knapp som har tryckts på, på knappsatsen.

Mikroprocessorns PD-anslutningar sätts till:

```
DDRD = 0b11000100;
```

Detta innebär att PD2, PD6 och PD7 används som utgångar. Portarna är anslutna till knappsatsens anslutningar som motsvarar dess rader.

En oändlig while-loop körs sedan.

```
while(1){
```

I loopens början sätts variabeln

```
int num = 1;
```

num används som argument vid anrop av funktionen blinkLed(num);

Sedan körs en for-loop ROWS antal gånger:

```
for(int j = 0; j < ROWS; j++){
```

I loopen sätts mikroprocessorns PD-utgångar PD2, PD7, PD6 sätts till 1 (hög/på) respektive.

```
PORTD = rowPin[j];
```

En nästlad for-loop kör COLS antal gånger:

```
for(int i = 0; i < COLS; i++){
```

I varje varv anropas funktionen checkPin, för att se om PD-ingången PD3, PD1, PD5 är hög, respektive. Om ingången är hög har en knapp tryckts in och blinkLed anropas med rätt siffervärde.

```
if(checkPin(columnPin[i])){
```

```
    blinkLed(num);
```

```
}
```

I varje varv ökas variabeln num med 1

```
num++;
```

Slut while-loop.

## 3 Bilagor

### 3.1 Programmetts källkod, main-funktion

```
#include <avr/io.h>

/* Defined in project properties */
// #define F_CPU 1000000ul
#include <util/delay.h>

#define ROWS 3 // 4
#define COLS 3

void blinkLed(int n);
int checkPin(int pin);

/* Row pins - for toggling output */
enum { ROW1_PIN = (0b00000100),
      ROW2_PIN = (0b10000000),
      ROW3_PIN = (0b01000000)
};

/* Column pins - for checking input */
enum { COL1_PIN = PD3,
      COL2_PIN = PD1,
      COL3_PIN = PD5
};

int main(void){
    DDRC = 0b00100000; //LED Output
    DDRD = 0b11000100; //Column pins output

    int columnPin[3] = {COL1_PIN, COL2_PIN, COL3_PIN};
    int rowPin[3] = {ROW1_PIN, ROW2_PIN, ROW3_PIN};

    while(1){
        int num = 1;
        /* Set row output */
        for(int j = 0; j < ROWS; j++){
            PORTD = rowPin[j];
            _delay_ms(10);
            /* Check column input */
            for(int i = 0; i < COLS; i++){
                if(checkPin(columnPin[i])){
                    blinkLed(num);
                }
                num++;
            }
        }
    }
}
```

### 3.2 Programmetts källkod, funktioner

```
/* Blink led n times */
void blinkLed(int n){
    for(int i = 0; i < n; i++){
        PORTC = 0b00100000;
        _delay_ms(200);
        PORTC = 0b00000000;
        _delay_ms(200);
    }
}

/* Check digital input on pin, returns 1 / 0 */
int checkPin(int pin){
    return (PIND & (1 << pin));
}
```