

Laboration 2: Tidmätning och synkronisering (ASCII-display)

Laboration 2 består av fyra uppgifter. De båda inledande uppgifterna behandlas utförligt i läroboken där de beskrivs i form av deluppgifter som succesivt måste klaras av innan själva laborationsuppgifterna kan lösas.

För att kunna genomföra laborationen måste du ha arbetat igenom avsnitten 5.1 t.o.m 5.3 i läroboken. Laborationsuppgifterna baseras på följande exempel och uppgifter:

- Exempel 5.1 och 5.2, uppgifter 5.1 t.o.m 5.9.

För en fördjupad förståelse kan du också studera dokumenten:

- Quick Guide: SysTick
- Datablad LCD – ASCII, sidor 12-16, 25, 28,29,33,34

Laborationsuppgift 2.1: Räknarmodul SysTick (Uppgift 5.3 i läroboken)

Uppgiften är att skapa de fördröjningsrutiner (realtid) som senare krävs för den synkroniserade kommunikationen med ASCII-displayen.

- Kontrollera programmets funktion med *CodeLite* och *SimServer*.
-

Laborationsuppgift 2.2: LCD – ASCII-display (Uppgift 5.9 i läroboken)

Uppgiften är att skapa funktioner för utmatning av text till ASCII-displayen.

- Kontrollera programmets funktion med *CodeLite* och *SimServer*.
-

Laborationsuppgift 2.3: ASCII-display och Keypad med textredigering

I denna uppgift kombinerar vi inmatningsrutinen `keyb_enhanced` från laboration 1 med ASCII-displayen för att skapa en applikation med enkel redigeringsfunktion.

Du får här ett exempel på hur man kan skapa ett numeriskt (0..9) tangentbord med ytterligare möjligheter som att radera ett inmatat tecken eller att avsluta applikationen.

I exemplet har vi implementerat funktionen på enklast tänkbara sätt och den har också minst en allvarlig svaghet: det görs ingen kontroll om insättningspunkten är giltig, dvs. inom ramen för ASCII-displayens positioner (1..20). Utöver de initieringar som krävs ska sådan kontroll också implementeras. Om försök till textinmatning utanför de tillåtna positionerna ska denna teckeninmatning ignoreras.

Laboration 2: Tidmätning och synkronisering (ASCII-display)

Delar av ett program som hanterar enkel textredigering visas till höger

Vi använder här ASCII-displayens rad 2 för textredigeringsfunktionen. En variabel `cursor_pos` anger insättningspunkten i raden.

För inmatning av tangentkoder 0-9 ska motsvarande ASCII-tecken skrivas i insättningspunkten.

Vid inmatning av tangentkod 0xB ska det sist inmatade tecknet raderas och insättningspunkten justeras.

Vid inmatning av tangentkod 0xD ska programmet avslutas.

```
cursor_pos = 1;
while( 1 )
{
    c = keyb_enhanced();
    switch( c )
    {
        case 0xD: return 0;
        case 0xB:
            cursor_pos --;
            ascii_gotoxy(cursor_pos,2);
            ascii_write_char( ' ' );
            ascii_gotoxy(cursor_pos,2);
            break;

        default:
            if( c < 10 )
            {
                ascii_write_char( c+'0' );
                cursor_pos ++;
            }
    }
}
```

- Skapa ett nytt projekt `ascii_keypad`.
- Implementera huvudprogrammet med kontrollfunktion för insättningspunkten
- Kontrollera programmets funktion med *CodeLite* och *SimServer*.

Laborationsuppgift 2.4: Inmatningsfunktion med samtidig visning

Funktionen i `ascii_keypad` kapslas i en funktion `get_command` och modifieras så att den redigerade textsträngen placeras i en parameter till funktionen och antalet inmatade tecken som placerats i textsträngen returneras från funktionen:

```
int get_command( char *string );
```

Eftersom det är en ASCII-textsträng som returneras ska strängen termineras med 0 innan den returneras.

Användning av funktionen `get_command` illustreras till höger.

Den anropande funktionen reserverar plats för textsträngen och skickar en pekare till `get_command`. Buffert för textsträngen är ett tecken större än ASCII-displayens utrymme för att även kunna rymma tecknet för terminering.

Om returvärdet är större än 0 betyder detta att minst ett tecken finns i teckensträngen.

```
char buffer[21];
...

if( get_command(buffer) )
{
    /* En textsträng finns i 'buffer' */
}
```

- Implementera `get_command` och ett enkelt huvudprogram för att testa funktionen.
- Kontrollera programmets funktion med *CodeLite* och *SimServer*.