

Arduinokurs



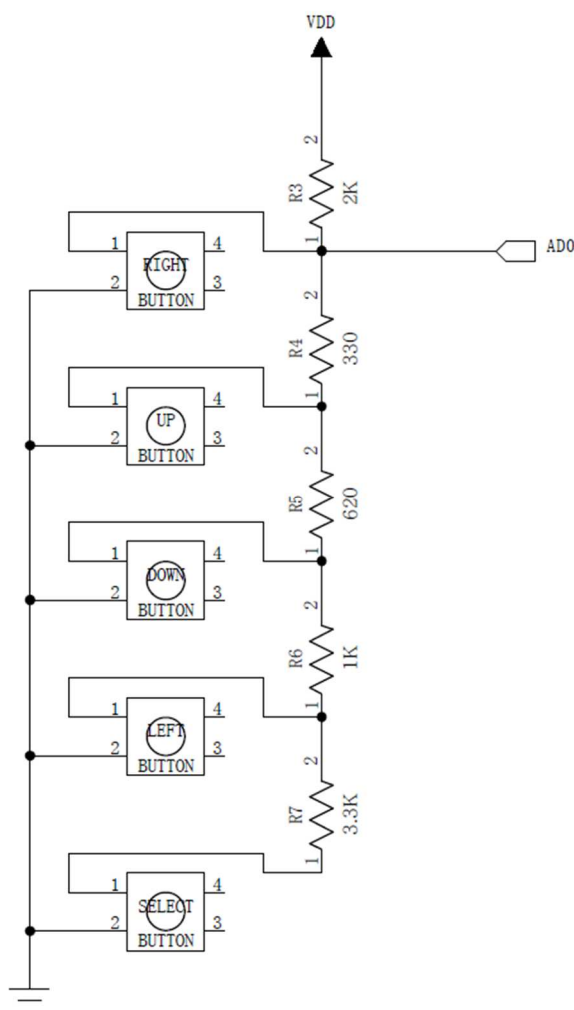
Kurstillfälle 2

Dagens lektion fortsätter där vi slutade sist, nämligen med AD-omvandlaren.

Vi följer till viss del lektion 3 i det utskickade materialet med undantag från att ansluta något annat än displayen till kortet. Alla kanske inte är så säkra på hur inkopplingen ska ske.

Displayen vi använder kommer från Electrokit 41010285 men även Kjell 87876 ska fungera. Jag refererar hela tiden till den förstnämnda.

Vi börjar med att titta på [schemat](#) och det som är intressant är tryckknapparna. Dessa är som vi ser kopplade till ADO – analogingång 0.



Arduinokurs

För att spara på portar har knapparna kopplats till en analogingång mha. en enkel spänningsdelare. Kopplingen är enligt mig lite konstig men det går vi inte in på nu. För att veta vilken knapp som är intryckt behövs lite enkel matematik samt beräkning av en spänningsdelare.

Vi tar knapparna var för sig.

"Select".

$$AD0 = \frac{5}{2000 + 330 + 620 + 1000 + 3300} (330 + 620 + 1000 + 3300) = 3,62V$$

"Left"

$$AD0 = \frac{5}{2000 + 330 + 620 + 1000} (330 + 620 + 1000) = 2,47V$$

"Down"

$$AD0 = \frac{5}{2000 + 330 + 620} (330 + 620) = 1,61V$$

"Up"

$$AD0 = \frac{5}{2000 + 330} (330) = 0,71V$$

"Right", AD0 = 0

Nu vet vi vilka olika spänningsnivåer som genereras när vi trycker på respektive knapp, men hur vet vi vilket värde i AD-omvandlaren som har motsvarande värde?

Kortets AD-omvandlare har 10 bitar och klarar en inspänning mellan 0 – 5V. Vi måste nu omvandla spänningen till AD-omvandlarens motsvarande värden 0 - 1023. Det största talet vi kan få med 10 bitar är $2^{10} - 1 = 1023$, men vi har 1024 tillstånd med nollan.

Eftersom vi har 5V och 1023 som största talet vid 10 bitar delar vi helt enkelt 5 med 1023.

$$\frac{5}{1023} = 4,89mV$$

Varje bit motsvarar alltså 5mV vilket också är upplösningen på våra mätningar.

För att få reda på AD-omvandlarens värde för respektive knapp räknar vi nu bakvägen fram de olika värdena genom att dividera "knappvärdet" med upplösningen.

$$\text{"Select"} = \frac{3,62}{4,89mV} = 741$$

$$\text{"Left"} = \frac{2,47}{4,89mV} = 505$$

Arduinokurs

$$\text{"Down"} = \frac{1,61}{4,89mV} = 329$$

$$\text{"Up"} = \frac{0,71}{4,89mV} = 145$$

Innan vi skriver programmet för att visa vilken knapp som är intryckt går vi igenom en ny funktion (statement) och den heter if (om).

Syntaxen för if ser ut så här:

```
if(värde)
{
  funktion;
}
```

Alltså om "värde" är sant (=1) utförs det inom klammerparenteserna/måsvingarna, annars inte. En viktig sak att komma ihåg är att "0" alltid är falskt och allt annat är sant!

Mjukvaran

Programmet som vi ska skriva / implementera kommer att skriva ut vilken knapp som har tryckts ned på serieporten och senare även på displayen.

När man ansluter något till en dator behövs oftast någon form av drivrutin. Till rutinen skickas kommandon och data som den i sin tur omvandlar till signaler och kommandon som den anslutna enheten förstår – i det här fallet displayen. Man kan naturligtvis skriva allt det här i sitt eget program men då krävs en hel del programmering och kunskap om hur displayen arbetar.

I C inkluderar man dessa sk. bibliotek (library) med kommandot #include. Mer om detta senare.

Implementering

En programmerare skriver inte kod – han implementerar ;-)

Hur gör vi nu för att visa vilken knapp som är nedtryckt?

Börja med att skriva in följande kod.

```
const int AD0 = A0;           //Deklarera AD0 (noll) till Analog in 0 (noll)
int AD0Value;                 //Deklarera AD0Value. Kommer att innehålla AD-värdet

void setup()
{
  Serial.begin(9600);         //Initiera seriell kommunikation
}
void loop()
{
  AD0Value = analogRead(AD0); //Läs AD-omvandlare och spara värde i AD0Value
  Serial.println(AD0Value);   //Skriv ut AD-värdet på serieporten och byter rad
  delay(1000);
}
```

Arduinokurs

Montera displayen om du inte har gjort det – se hårdvara på nästa sida!

Ladda upp koden och tryck sedan på 'Verktyg' – 'Seriell monitor'. Verifiera att knapparna fungerar och jämför värdena.

Nästa steg blir att skriva ut vilken knapp som är nedtryckt. Hur görs det?

Jo, genom villkor. Vi har värdena 741, 505, 329, 145 och 0.

Om värde = 741 skriver vi "Select"

Om värde = 505 skriver vi "Left"

Om värde = 329 skriver vi "Down"

Om värde = 145 skriver vi "Up"

Om värde = 0 skriver vi "Right"

Vi skulle kunna skriva nedanstående men det skulle till 99% inte fungera. Varför?

Svaret är toleranser i bl.a. motståndsvärdena. Vi räknar inte på det nu, men ha det i åtanke!

```
if(button == 741)
{
  lcd.print("Select");      //Skriv "Select"
}
```

```
if(button == 505)
{
  lcd.print("Left");       //Skriv "Left"
}
```

OBS mycket viktigt!

I C är det viktigt att skilja på tilldelning och jämförelse. "=" är tilldelning och "==" är lika med.

Exempel:

A = 10 //A tilldelas värdet 10

B = A //B tilldelas A vilket är 10

C = 5 //C tilldelas värdet 5

if(A=B){} //A tilldelas B. Satsen är sann om B inte är noll

if(A==B){} //Om A är lika med B. Satsen sann om A är lika med B

Vi kan nu skriva följande kod i void loop()

```
void loop()
{
  AD0Value = analogRead(AD0); //Läs AD-omvandlare och spara värde i AD0Value
  if(AD0Value == 741) //Är AD0 lika med 741?
  {
    Serial.println("Select"); //Om ja Skriv ut Select på serieporten
  }
  delay(1000);
}
```

Den här koden kommer att fungera dåligt pga just toleranser på komponentvärdena.

Om vi lägger till else if ('annars'-funktionen) blir det bättre.

Arduinokurs

```
void loop()
{
  AD0Value = analogRead(AD0);      //Läs AD-omvandlare och spara värde i AD0Value
  if(AD0Value < 100)               //Är AD0 mindre än 100?
  {
    Serial.print("Right");         //Om ja Skriv ut Right på serieporten
  }
  else if(AD0Value < 200)          //Vi lägger oss en bit ovanför 145
  {
    Serial.print("Up");           //Om ja Skriv ut Up på serieporten
  }
  delay(1000);
}
```

Fundera nu ett tag på hur övriga knappar ska skrivas ut. Det finns flera sätt att lösa uppgiften!

Hårdvaran

Koppla loss USB-kabeln innan du fortsätter!!

Montera displayen om du inte redan ha gjort det och tryck försiktigt fast den som på bilden.

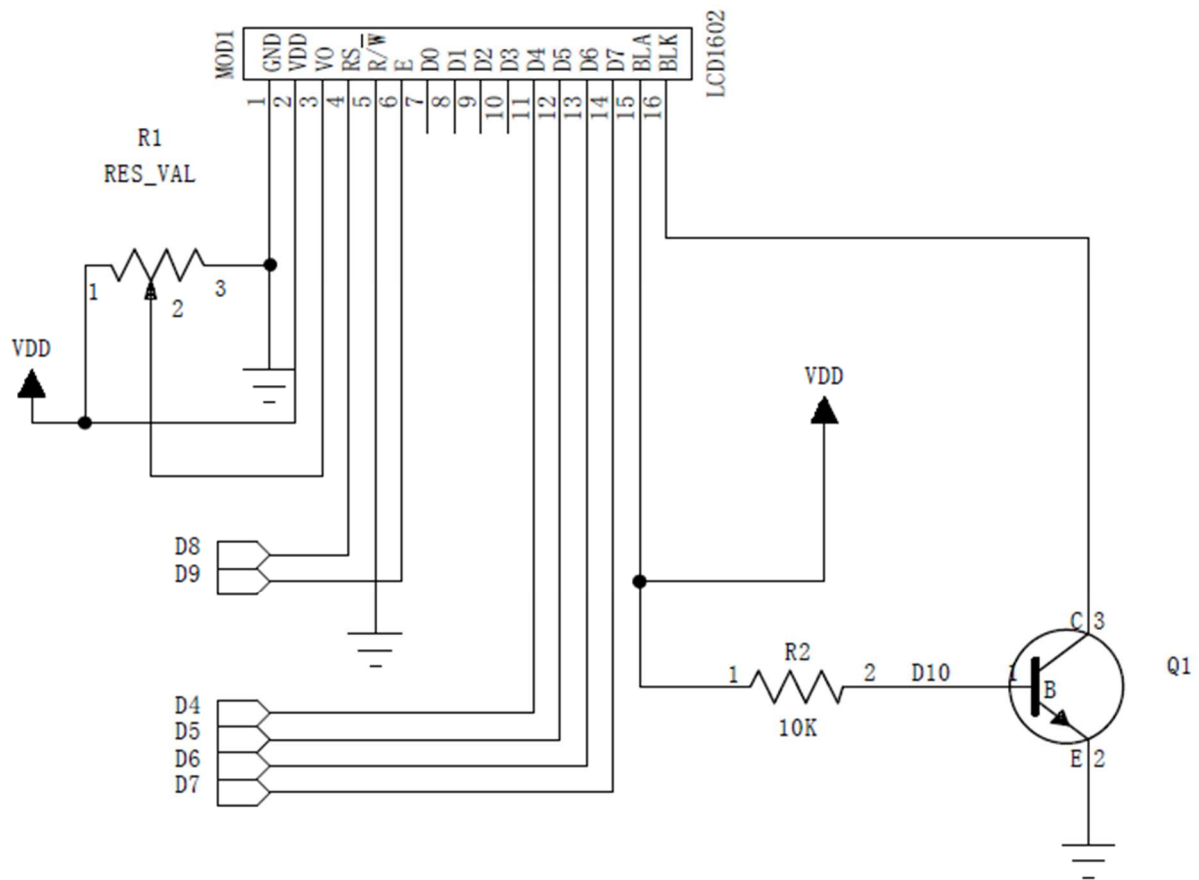
De vänstra stiften på displayen längst till vänster på Arduinon.



Vi tar en titt på schemat igen. Displayen har ett flertal anslutningar och dessa är kopplade till ett antal portpinnar. Var och en med sin egen funktion. I programmet måste vi tala om vilken funktion som används till vilken portpinne. Mer om det vid implementeringen men vi tar ett exempel.

Längst upp i schemat har vi displayen LCD1602 (16 tecken och 2 rader). Pin 4 heter "RS" och den är kopplad till "D8". Detta kommer vi ihåg till koden.

Arduinokurs



```
#include <LiquidCrystal.h> //Inkluderar funktioner för displayen
```

```
#define RSPin 8           //RS till D8  
#define EnablePin 9     //E till D9  
#define DS4 4           //D4 till D4  
#define DS5 5           //D5 till D5  
#define DS6 6           //D6 till D6  
#define DS7 7           //D7 till D7
```

```
#define LCDColumns 16   //Displayen har 16 tecken per rad  
#define LCDRows 2      //Displayen har 2 rader
```

```
const int AD0 = A0;     //Deklarera AD0 (noll) till Analog in 0 (noll)  
int AD0Value;          //Deklarera AD0Value. Kommer att innehålla AD-värdet
```

```
LiquidCrystal lcd(RSPin, EnablePin,DS4,DS5,DS6,DS7); //Initierar portpinnarna
```

```
void setup()  
{  
  lcd.begin(LCDColumns, LCDRows); //Configurerar LCDn  
  lcd.setCursor(0,0);             //Sätter markören till övre vänstra hörnet  
  lcd.print("Hello world");      //Skriv det klassiska "Hello world"
```

Arduinokurs

```
}  
  
void loop()  
{  
}
```

Nu kan vi prova vår föregående kod om vi ändrar utskriften från serieporten till displayen.

```
void loop()  
{  
  AD0Value = analogRead(AD0); //Läs AD-omvandlare och spara värde i AD0Value  
  if(AD0Value < 100)          //Är AD0 mindre än 100?  
  {  
    lcd.clear();              //Raderar all text och sätter markören till övre vänstra hörnet  
    lcd.print("Right");       //Om ja Skriv ut "Right" på displayen  
  }  
  else if(AD0Value < 200)  
  {  
    lcd.clear();              //Raderar all text och sätter markören till övre vänstra hörnet  
    lcd.print("Up");          //Om ja Skriv ut "Up" på displayen  
  }  
  delay(500);  
}
```

Avslutningsvis visas ett alternativt sätt att implementera det hela på. Det sker med jämförelse samt en logisk och-funktion. Vi kommer att gå igenom logiskt-och samt bitvis-och framöver. Och i det här sammanhanget skrivs && och betyder att båda funktionerna måste vara sanna för att funktionen ska utföras.

```
void loop()  
{  
  AD0Value = analogRead(AD0); //Läs AD-omvandlare och spara värde i AD0Value  
  if(AD0Value < 100)          //Är AD0 mindre än 100?  
  {  
    lcd.clear();              //Raderar all text och sätter markören till övre vänstra hörnet  
    lcd.print("Right");       //Om ja Skriv ut "Right" på displayen  
  }  
  
  //OM AD0Value är både större än 100 OCH mindre än 200  
  if((AD0Value > 100) && (AD0Value < 200))          //Observera parenteserna  
  {  
    lcd.clear();              //Raderar all text och sätter markören till övre vänstra hörnet  
    lcd.print("Up");          //Om ja Skriv ut "Up" på displayen  
  }  
  delay(500);  
}
```

Skriv koden för de återstående knapparna...