

M

odelljärnvägs- magasinet

Bilaga till nummer 18

Mikrodatorer på modelljärnväg

Utökad stationsstyrning med Arduino

Av Lars Ljungberg

Utfarten västerut från Kils station – nog så komplex att styra. Foto: Stefan Nilsson.



I nummer 17 beskrev jag hur man kan styra en station med hjälp av Arduino. Sedan dess har läsare hört av sig läsare med behov av både fler knappar och växlar, så här är en utökad version.

Vad som skiljer • Det som framför allt skiljer den ursprungliga och denna version är att denna använder sig av MCP-kretsar till att läsa av knappar, strömsensorer och styra reläer för växelomläggning.

Det innebär att man kan bygga ut antalet efter behov och är därmed inte begränsad till de in- och utgångar som Arduino UNO har tillgängliga.

Dessutom har jag lagt till en ny signaltyp: dvärgsignalen.

Avläsning av knappar • För knapparna har jag skapat en ny klass, `Button`, för att hantera avläsningen via MCP.

Vid initiering i setup skickar man med följande parametrar:

McpId: Adress (0-7) för den MCP-krets som man kopplat sin knapp till.

IO: Till vilken IO-port (1-16) som knappen är ansluten.

ButtonId: Vilket ID man skall få tillbaka när knappen är nedtryckt.

En liten finess med `Button`-klassen är att den själv kan konfigurera om vald IO till ingång samt aktiverar det inbyggda pullup-motståndet. Det innebär att du inte behöver ha några pullup-motstånd ihop

med knapparna, utan ingången jordas när en knapp trycks ner (se figur 1).

Dessutom kan du, om behovet finns, utnyttja oanvända IO på andra MCP, till exempel en MCP som annars bara styr signaler och slippa att lägga till ytterligare en MCP bara för att det saknas en ingång på den befintliga MCP:n som används för knappavläsning.

Strömsensorer • Klassen för strömsensorer, `Detector`, har fått en liten justering så att de kopplas via MCP istället.

Vid initiering i setup skickar man med följande parametrar:

McpId: Adress (0-7) för den MCP-krets som strömsensorn är kopplad till.

IO: Till vilken IO-port (1–16) som strömsensorn är ansluten.

Strömsensorerna har samma finess som knapparna, det vill säga att de kan konfigureras till valfri ledig IO på valfri MCP.

Styrning av växlar • När det gäller styrning av växlar har klassen Turnout justerats så att följande parametrar anges vid initiering i setup:

McpId: Adress (0-7) för den MCP-krets som reläet är anslutet till.

IO: Vilken IO port (1-16) som reläet är anslutet till.

DelayTime: Tid i millisekunder för fördröjning så att växeln hinner slå om

UsePulse: Flagga; om den är sann (true) skickas en puls via pulsreläet vid omläggning av denna växel. Används för växlar som har ”elektromagnetmotor”.

Pulsreläet styrs via en utgång direkt från Arduino för att spara IO på MCP-kretsarna.

Portarna i en MCP klarar inte att driva så mycket, så reläerna förutsätts ha transistordrivsteg samt diodskydd över spolen. Figur 2 visar hur räler och pulsrelä kopplas in.

Dvärgsignaler • Bygger du exempelvis en liten bangård kan behovet av dvärgsignaler finnas. De förbrukar fyra IO per signal. I vilken ordning ”lamporna” (lysdioderna) skall vara anslutna framgår av figur 3.

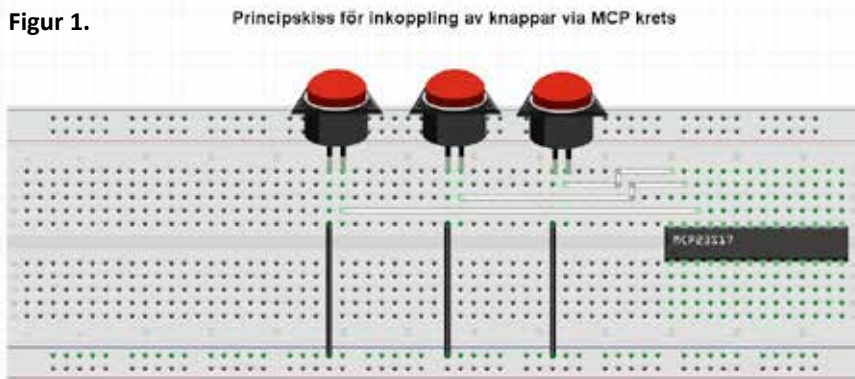
I verkligheten går en dvärgsignal över till att visa varsamhet (45 grader) när den har passerats av första fordonet men det fortfarande finns fordon kvar på spårledningen före signalen. Detta simulerar jag genom att låta signalen visa varsamhet några sekunder innan den övergår till stopp, om den sätts till stopp via en strömsensor. Eftersom alla inte har spårledning och vagnar med motståndsaxlar anser jag det är en bra kompromiss jämfört med att slå om direkt till stopp.

Optimering • Du kan således blanda flera funktioner på samma MCP, bara du är medveten om hur saker och ting fungerar och inte försöker använda samma pinne till flera saker.

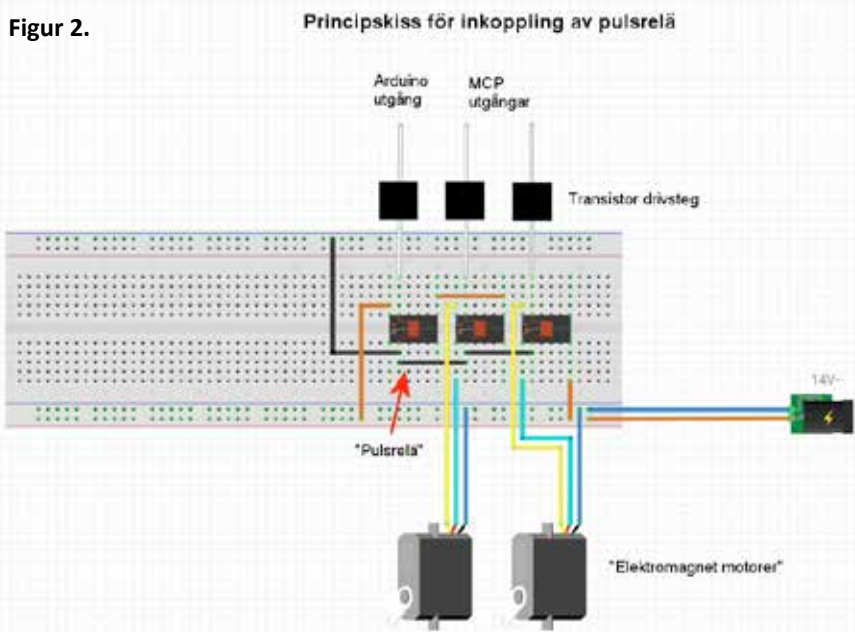
Det är alltså inte tvunget att använda en MCP per funktion, även om det i vissa fall kan vara praktiskt. Till exempel använder jag samma MCP både för knapparna och strömsensorer i exempelkoden.

Det är inget som hindrar att man exempelvis har både signal- och växelstyrning samt knappavläsning via samma MCP.

Figur 1.

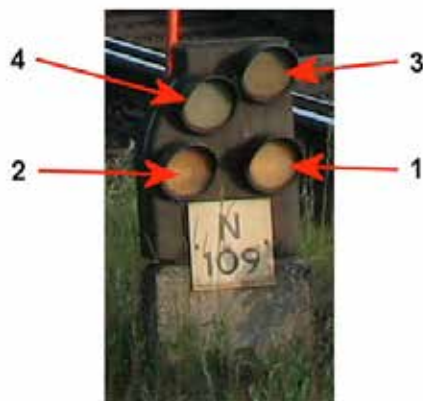


Figur 2.



Figur 3.

Ordning för ”lamporna” i dvärgsignalen



Extra spänningsmatning • När antalet signaler och reläer ökar krävs en extra stabiliserad spänningskälla på 5 V, då Arduino Uno max ger 0,5 A.

Det kan också vara lämpligt att mata alla MCP från den externa spänningskällan, och bara koppla ihop GND (jord) från spänningskällan till GND på Arduino Uno.

Nästa steg? • Ytterligare en tänkbar utökning vore att lägga till möjlighet för fler än en samtidig tågväg. För en bangård av lite annan karaktär än mitt exempel vore det mycket praktiskt.

Vi får se vad framtida nummer MJ-magasinet kan ha att erbjuda. ❀