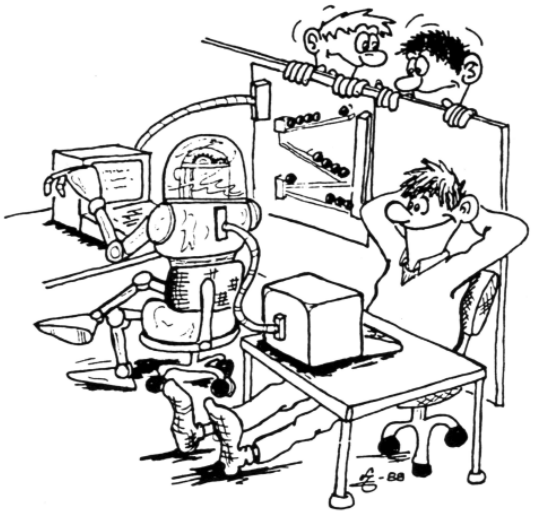
# Industriautomation

**Tentamen SSY 066**, tisdag 22/04 -2014, em, V

Lärare: Kristofer Bengtsson, 0768 979561



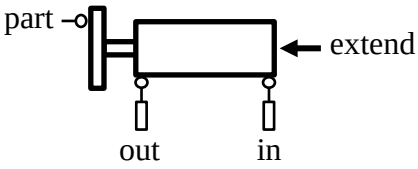
Fullständig lösning ska lämnas på samtliga uppgifter. I förekommande fall av tvetydigt formulerade tentamensuppgifter ska den föreslagna lösningen och eventuella antaganden motiveras. Examinator förbehåller sig rätten att godkänna rimligheten i antaganden och motiveringar.

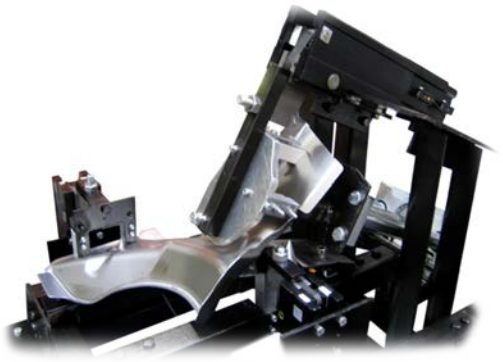
Totalt omfattar tentamen 15 poäng. För betygen tre, fyra och fem krävs 7, 10 resp 13 poäng. Lösningar anslås första vardagen efter tentatillfället på kursens hemsida i Pingpong. Granskning av rättningen sker 28 april kl. 12:30 – 13:30 på institutionen.

OBS. Inga hjälpmedel är tillåtna.

# Uppgift 1

I denna uppgift skall vi styra en cylinder i fixturen till höger.
En förenklad bild av cylindern är:



Cylindern har två lägesgivare, insignalerna ***in*** och ***out*** som säger om cylindern är inne (***in***=true) eller ute (***out***=true). Den styrs av utsignalen ***extend***. Så länge ***extend*** är true kommer cylindern att röra sig ut och stanna ute. För att föra tillbaka cylindern skall ***extend*** sättas till false.

Det finns också en sensor ***part*** som är true om en plåtbit finns på cylinderns platta.

**a)** Implementera EN ladder rung som styr utsignalen ***extend***. Den skall aktiveras då er kod får styrsignalen ***move_out*** från ett annat styrsystem. Så länge ***move_out*** är true så skall ***extend*** vara true. Om ***move_out*** ändras till false skall också ***extend*** bli false förutom om det finns en plåtbit som aktiverar sensorn ***part***. Om detta är fallet skall cylindern inte åka in, förrän plåtbiten försvinner.

(2 poäng)

**b)** En utmaning när vi styr cylindrar är att hantera om något blir fel. I denna uppgift skall ni implementera ett larm som varnar om cylindern tar för lång tid på sig att komma helt ut. Alltså tiden det tar innan sensorn ***out*** blir true.

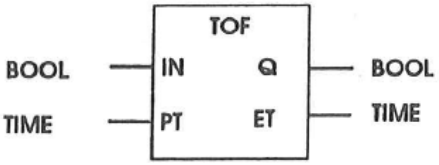
Om det tar mer än 2 sekunder (T#2s) från utsignal tills cylindern är ute, så skall signalen ***alarm*** sättas till true. Om ***alarm*** blir true så skall den fortsättningsvis vara true. Så er kod skall alltså inte återställa den. Koden skall vara skriven i ladder logic.

Olika timerfunktionsblock är beskrivna på nästa sida.

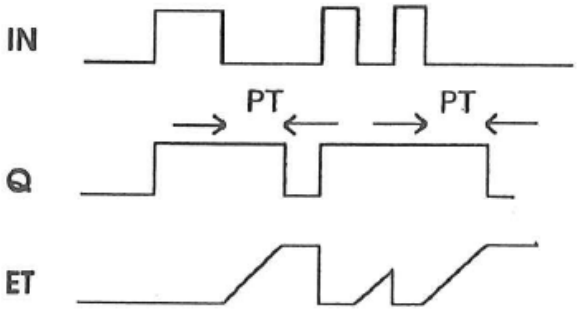
(2 poäng)

# Timers

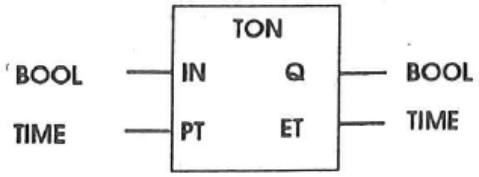
Off Timer function block



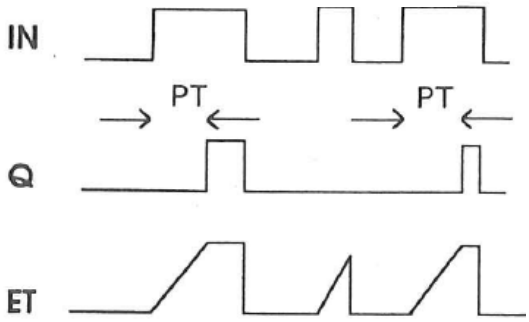

Off Timer timing diagram



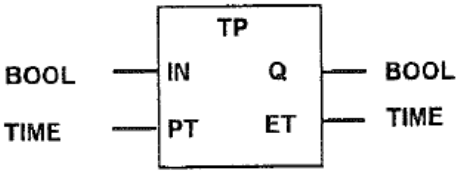

On Timer function block



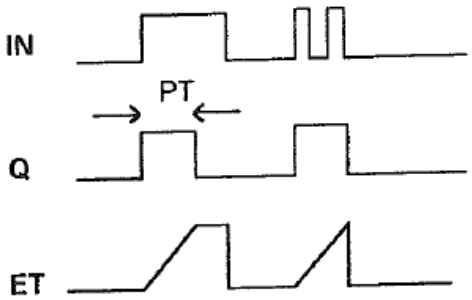

On Timer timing diagram



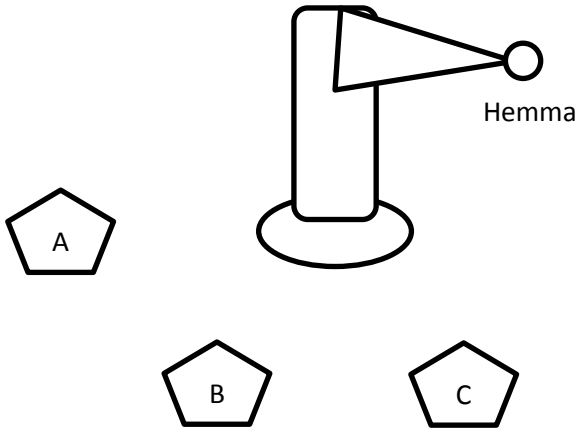

Puls Timer function block




Puls Timer timing diagram



# Uppgift 2

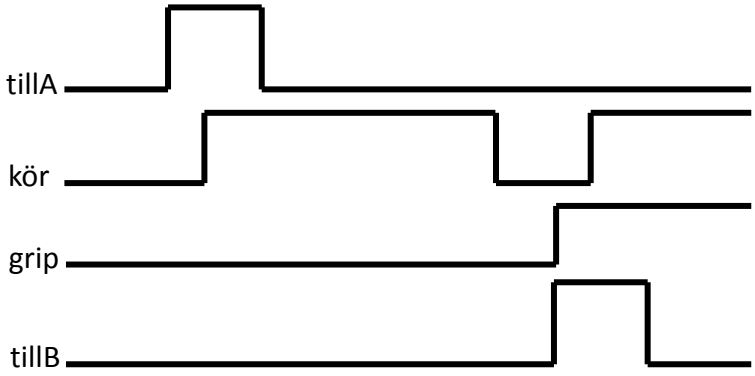



I uppgift 2 och 3 skall en robot styras så att den kan plocka upp och lämna produkter vid tre positioner, A, B och C. I denna uppgift skall delar av den detaljerade PLC-styrningen implementeras. Du skall implementera styrningen så att roboten hämtar en produkt vid A och lämnar den vid B med hjälp av ladder rungs.

Roboten styrs av fem boolska PLC-utsignaler: *tillA, tillB, tillC, hem* och *grip*. Roboten skickar en boolsk insignal till PLCn, *kör*, som är sann då roboten rör på sig.

*tillA, tillB, tillC* och *hem* får inte vara sanna samtidigt. För att styra roboten så att den åker till A, skall signalen *tillA* sättas sann. Efter en liten stund kommer roboten skicka tillbaka signalen *kör*. När det sker skall din kod sätta *tillA* falsk. När sedan *kör* blir falsk så är roboten framme vid A. Då skall din kod plocka produkten genom att sätta *grip* sann. *Grip* skall vara sann fram tills roboten skall släppa produkten. Roboten griper och släpper produkten snabbt så tiden för det behöver inte hanteras av koden.

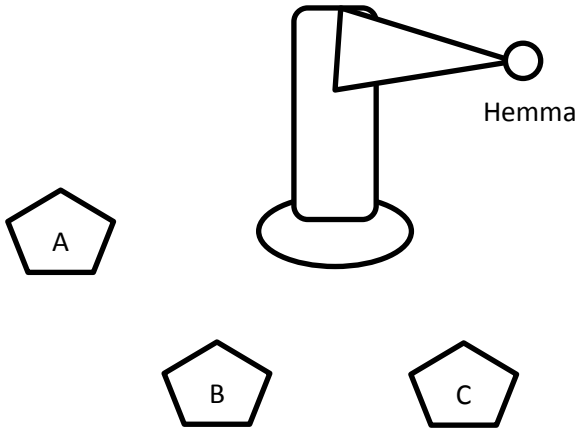
Början på ett tidsdiagram för din styrningen är:




Implementera ladder rungs som plockar upp en produkt vid A och lämnar den vid B.

(6 poäng)

# Uppgift 3




När den detaljerade styrningen är implementerad kan roboten styras med hjälp av signalerna *AB, AC, BA, BC, CA, CB.* När en signal, till exempel *AB,* skickas till roboten åker den till A, hämtar en produkt och lämnar den vid B och åker sedan till hemmaläget. I denna uppgift tar robotens förflyttning ingen tid. Så det räcker med att skicka signalen till roboten, så är det utfört. (Använd true som övergångsvilkor efter strysignalen i SFCn)

I denna uppgift skall du implementera en SFC som flyttar runt två olika produktyper, $P_1$ och $P_2$ mellan positionerna efter ett visst recept. Vid position B och C utför operatörer en viss uppgift på produkten. När operatören är färdig trycker den på en knapp som skickar signalerna *PosBFärdig* och *PosCFärdig*. När dessa är sanna skall din kod flytta produkten till nästa position.

När produkten $P_1$ läggs vid A, vilket signaleras med signalen *P1VidA* skall den först flyttas till position C. När den är färdig, vilket signaleras av *PosCFärdig,* flyttas den till B och efter *PosBFärdig,* sedan till A. När den hamnar vid A så tas den bort av en operatör direkt och är färdig. Efter det kan en ny produkt av typ $P_1$ bearbetas av systemet.

Produkt $P_2$ skickar signalen *P2VidA* och skall sedan flyttas till B, sedan till C och slutligen till A.

Produkterna kan vara på samma position samtidigt, men vid signalerna *PosBFärdig* eller *PosCFärdig* så är det produkten som kom till positionen först som är färdig. Det är allstå en FIFO buffert. Så om $P_1$ kommer till position B när $P_2$ redan är där så gäller första *PosBFärdig* $P_2$ vilket betyder att roboten skall exekvera *BC*. Efter nästa *PosBFärdig* skall *BA* exekveras för att flytta $P_1$ till A.

Implementera en SFC som kan flytta runt produkterna i cellen enligt beskrivningen ovan. Koden skall kunna hantera att båda produkterna är samtidigt i cellen i vilken ordning som helst, även en åt gången.

Tips: Lös först koden för att flytta runt en produkt i taget. Fundera sedan vad som behöver ändras och / eller läggas till för att hantera båda produkterna samtidigt.

(5 poäng)