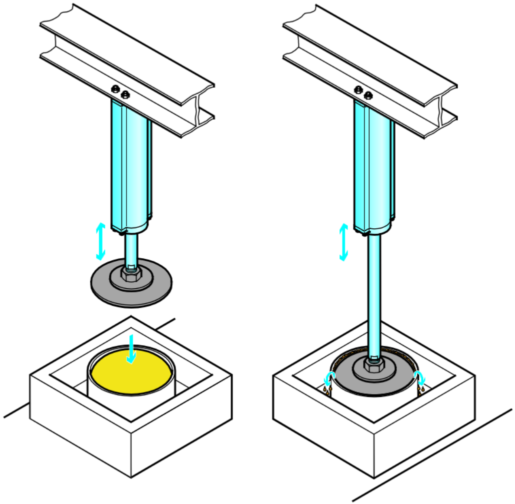
Logic control of electro-pneumatic systems

Kjartan Halvorsen

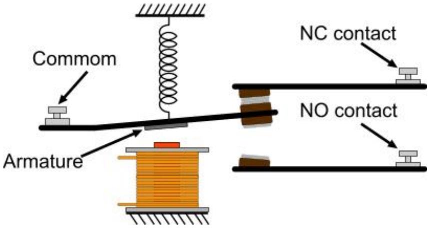
November 4, 2021

Cheese pressing example, sequence A+A-

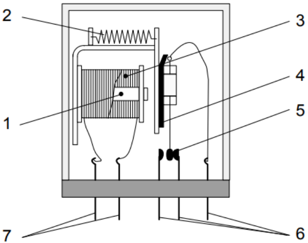


From FESTO Didactic

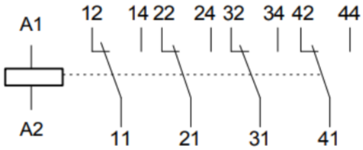
The Relay



From pcbheaven.com



From FESTO didactic



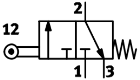
From FESTO didactic

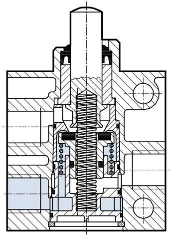
From FESTO didactic

Other key components

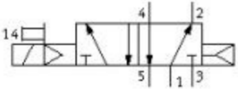
Sources: FESTO didactic, electroschematics.com, automation-insights.blog

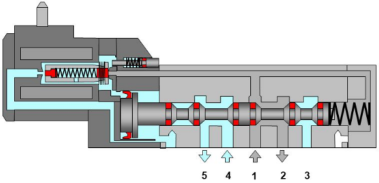
Limit switch



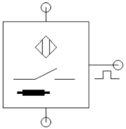


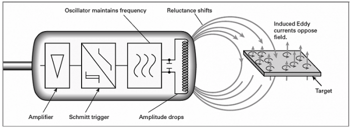
Solenoid valve

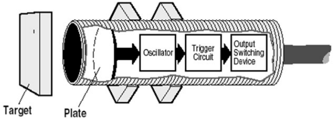




Proximity sensor

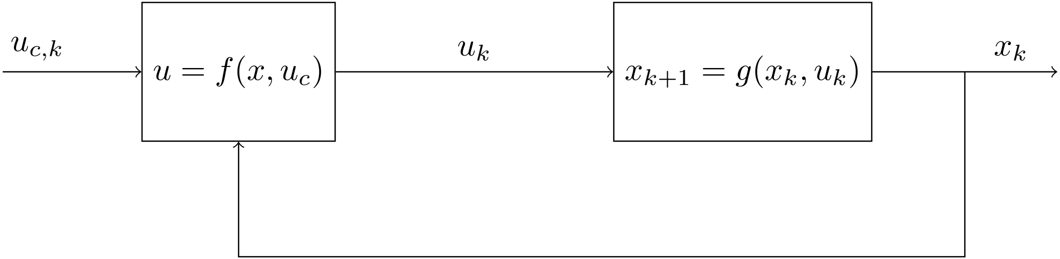






A logic control loop

controller = logic circuit plant = pneumatic system



Cheese pressing example - Variables

State variables

$x = \begin{bmatrix} x_R & x_E \end{bmatrix}^T$ with

$$x_R = \begin{cases} 1 & \text{Cylinder retracted} \\ 0 & \text{not retracted} \end{cases}$$

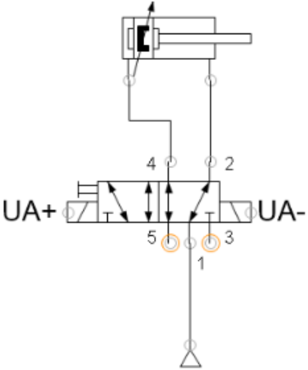
$$x_E = \begin{cases} 1 & \text{Cylinder extended} \\ 0 & \text{not extended} \end{cases}$$

Control signal

$u = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T$, with

$$u_1 = \begin{cases} 1 & \text{Activate UA+} \\ 0 & \text{Don't activate UA+} \end{cases}$$

$$u_2 = \begin{cases} 1 & \text{Activate UA-} \\ 0 & \text{Don't activate UA-} \end{cases}$$



Activating solenoid UA+ extends the cylinder, activating UA- retracts the cylinder.

Command signal

$$u_c = \begin{cases} 0 & \text{Button unpushed} \\ 1 & \text{Button pushed} \end{cases}.$$

Cheese pressing example - Plant dynamics

Plant dynamics $x_{k+1} = g(x_k, u_k)$

Input		Current state		Next state	
$u_{1,k}$	$u_{2,k}$	$x_{R,k}$	$x_{E,k}$	$x_{R,k+1}$	$x_{E,k+1}$
0	0	0	1	0	1
0	1	0	1	1	0
1	0	0	1	0	1
(1)	(1)	(0)	(1)	(0)	(1)
0	0	1	0	1	0
0	1	1	0	1	0
1	0	1	0	0	1
(1)	(1)	(1)	(0)	(1)	(0)

Intermezzo - Maxterms and minterms

Minterms

A minterm is a boolean expression that is TRUE (=1) for one and only one row in the truth table. For instance $Y = X_1X_2X_3$ will only be true when $X_1 = X_2 = X_3 = 1$, and $Y = \overline{X_1}X_2\overline{X_3}$ will only be true if $X_1 = X_3 = 0$ and $X_2 = 1$. The combination $Y = X_1X_2X_3 + \overline{X_1}X_2\overline{X_3}$ will have only two rows equal to 1 in the truth table.

Example:

Inputs			Outputs	
X_1	X_2	X_3	Y_1	Y_2
0	0	0	0	1
0	0	1	0	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0	1

$Y_1 = m_2 + m_3 = \overline{X_1}X_2\overline{X_3} + \overline{X_1}X_2X_3, \qquad Y_2 =$

Maxterms

A maxterm is a boolean expression that is FALSE (=0) for one and only one row in the truth table. For instance $Y = X_1 + X_2 + X_3$ will only be false when $X_1 = X_2 = X_3 = 0$, and $Y = \overline{X_1} + X_2 + \overline{X_3}$ will only be false if $X_1 = X_3 = 1$ and $X_2 = 0$. The combination $Y = (X_1 + X_2 + X_3)(\overline{X_1} + X_2 + \overline{X_3})$ will have only two rows equal to 0 in the truth table.

Example:

Inputs			Outputs	
X_1	X_2	X_3	Y_1	Y_2
0	0	0	0	1
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0

$Y_1 = M_0 M_1 = (X_1 + X_2 + X_3)(X_1 + X_2 + \overline{X_3})$, $\qquad Y_2 =$

Cheese pressing example - Control law

The system is operating as long as the start button is pressed ($u_c = 1$). When the button is released, the cylinder should go to the retracted position.

Control law $u_k = f(x, u_c)$

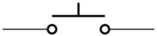
x_R	x_E	u_c	u_1	u_2
0	1	0	0	1
1	0	0	0	0
0	1	1	0	1
1	0	1	1	0
0	0	0	0	1
0	0	1	0	0

Activity: Write as boolen functions

$$u_1 = f_1(x_R, x_E, u_c) =$$
$$u_2 = f_2(x_R, x_E, u_c) =$$

Cheese pressing example - implementing the control law

normally open



normally closed

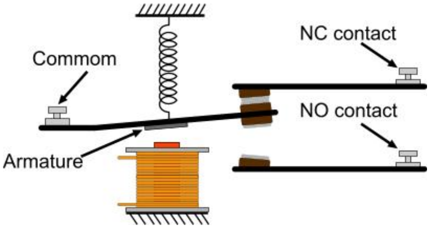
normally open

normally closed

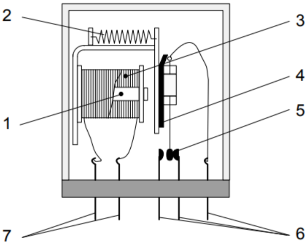
normally open

normally closed

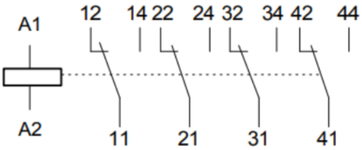
Intermezzo - An electrical circuit with memory



From pcbheaven.com



From FESTO didactic

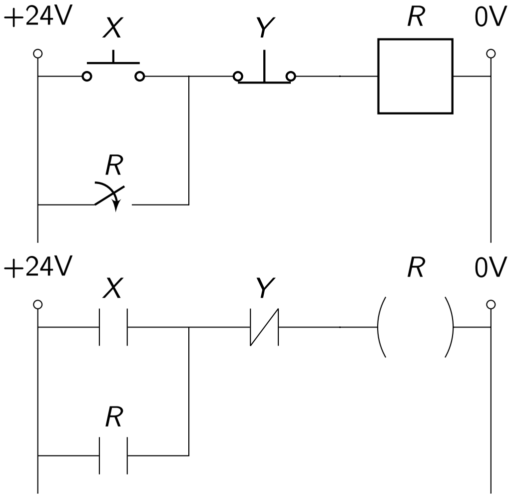


From FESTO didactic

From FESTO didactic

Intermezzo - An electrical circuit with memory

Latching circuit

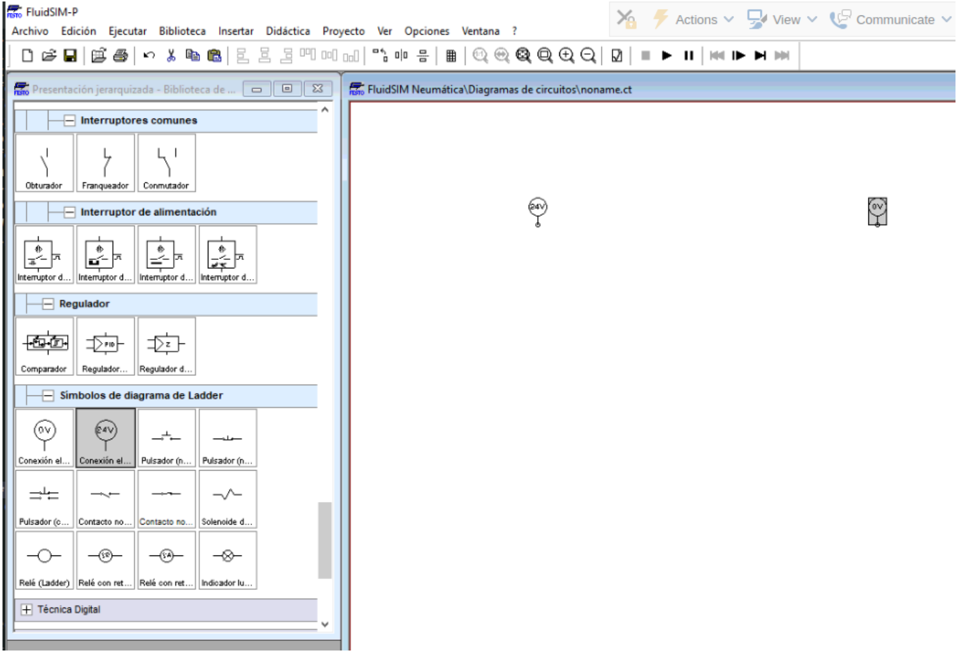


Truth table

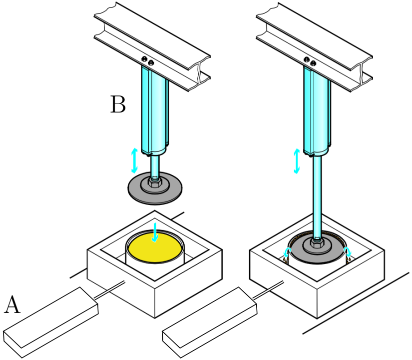
X	Y	R_k	R_{k+1}
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

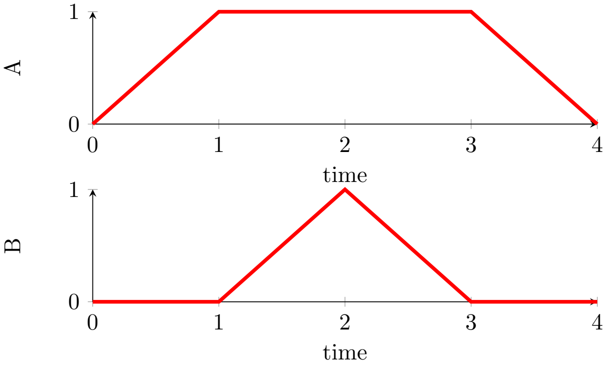
Group activity: Implement the circuit in FluidSim and verify the truth table.

Electrical circuits in FluidSim



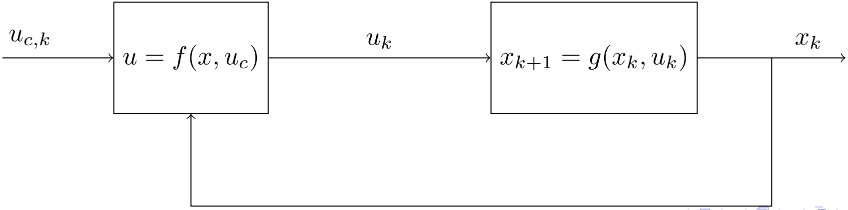
The lab assignment



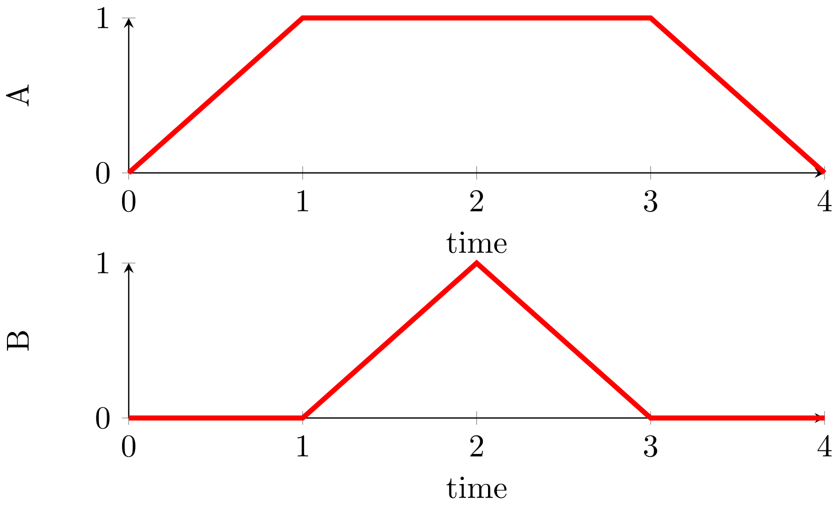


controller = logic circuit

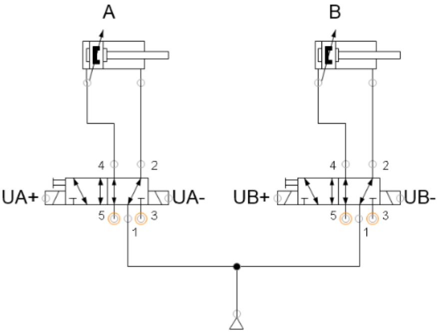
plant = pneumatic system



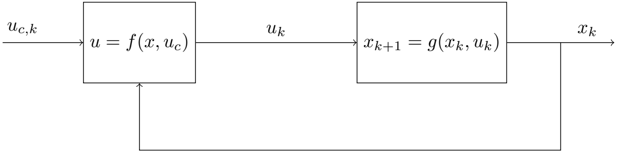
Implementing the sequence A+B+B-A-



Implementing the sequence A+B+B-A-, control signal



controller = logic circuit   plant = pneumatic system

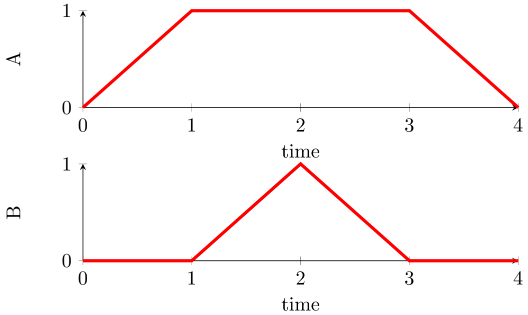


Control signal

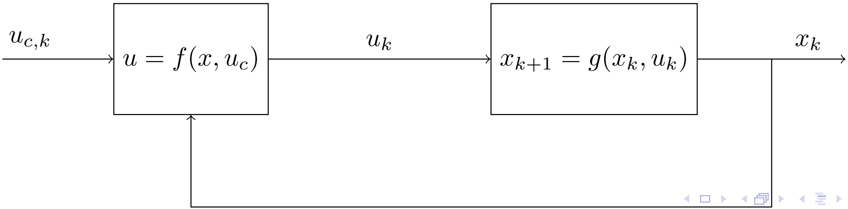
$$u = \begin{bmatrix} u_{A}+ & u_{A}- & u_{B}+ & u_{B}- \end{bmatrix}^T,$$

Implementing the sequence A+B+B-A-, the problem

The correct control signal (action) is not uniquely defined by the position of the cylinders



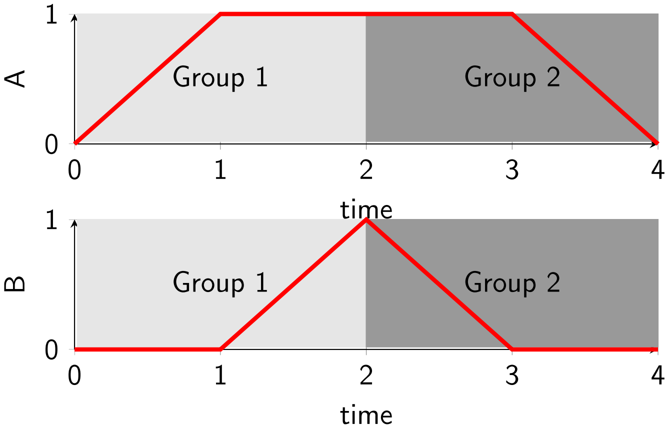
controller = logic circuit   plant = pneumatic system



Implementing the sequence A+B+|B-A-

Dividing the sequence into groups (a.k.a. cascade method) Each group contains as many steps as possible without repeating a letter.

$$\underbrace{A+B+}_{\text{Group 1}} \mid \underbrace{B\text{-}A\text{-}}_{\text{Group 2}}$$

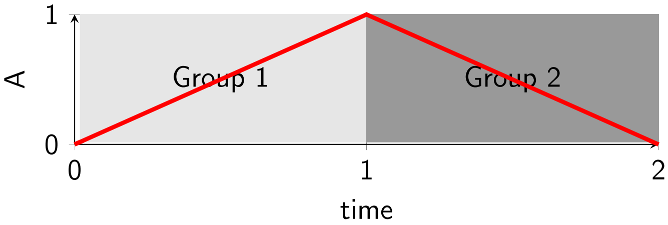


The cascade method applied to A+A-

The cascade method applied to A+A-

Divide the sequence is to groups, where each group is as long as possible without repeating the same letter.

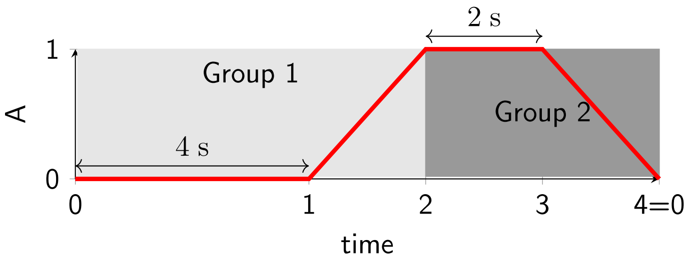
$$\underbrace{\text{A+}}_{\text{Group 1}} \mid \underbrace{\text{A-}}_{\text{Group 2}}$$



The cascade method applied to A+A- with delays

Let's add some delays. The process is cyclic and automatic. It takes 4 seconds to replace the mold under the press. The cheese needs to be pressed during 2 seconds before the cylinder retracts.

$$\underbrace{T_{4s}\ \text{A+}}_{\text{Group 1}} \mid \underbrace{T_{2s}\ \text{A-}}_{\text{Group 2}}$$



State variables

State variables

$$x = \begin{bmatrix} x_R & x_E & x_{G1} & x_{G2} & x_{T4} & x_{T2} \end{bmatrix}^T,$$

where

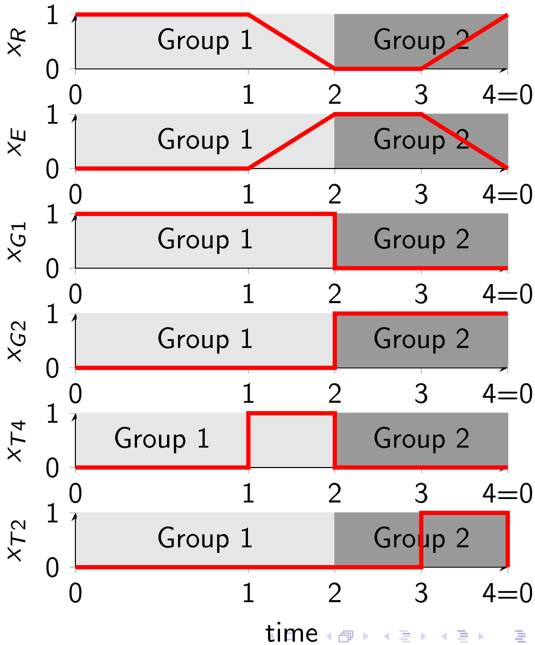
$$x_R = \begin{cases} 1 & \text{Cylinder A retracted} \\ 0 & \text{not retracted} \end{cases}$$

$$x_E = \begin{cases} 1 & \text{Cylinder A extended} \\ 0 & \text{not extended} \end{cases}$$

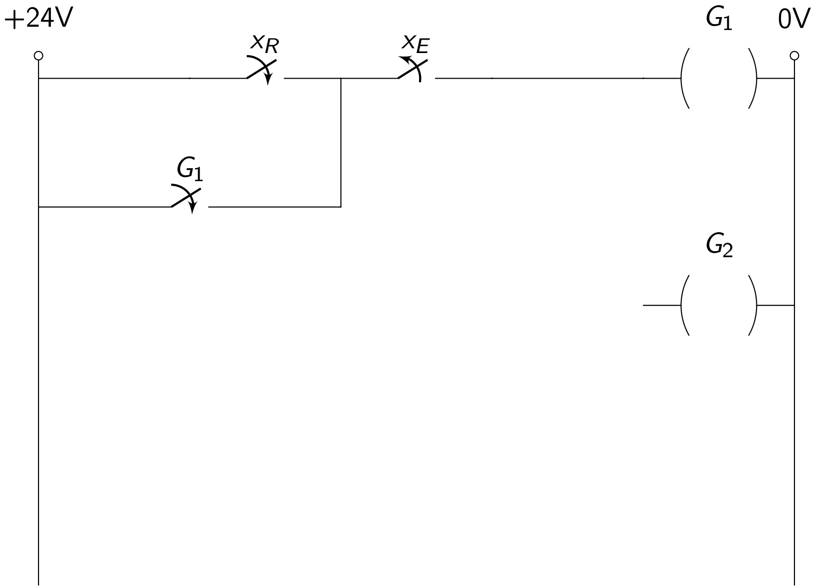
$$x_{Gi} = \begin{cases} 1 & \text{Group } i \text{ active} \\ 0 & \text{Group } i \text{ not active} \end{cases}$$

$$x_{Ti} = \begin{cases} 1 & \text{Timer of } i \text{ s completed} \\ 1 & \text{Timer of } i \text{s not completed} \end{cases}$$

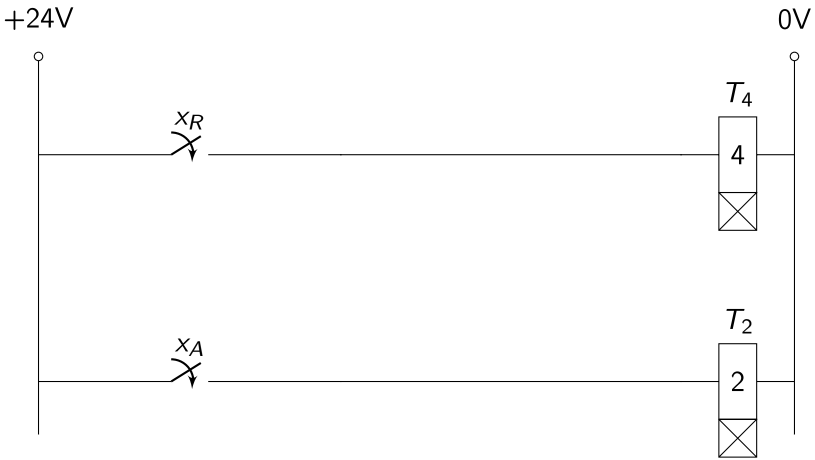
State transitions



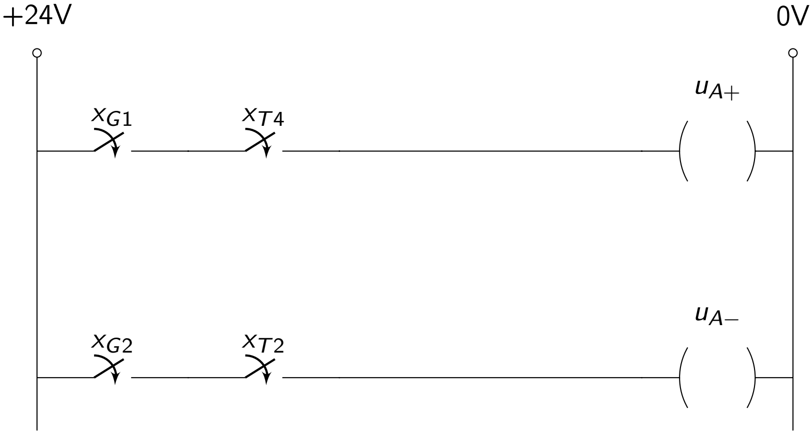
Group transitions



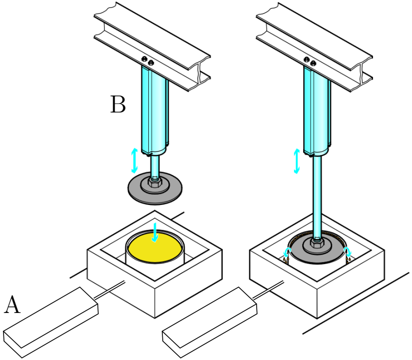
The timers

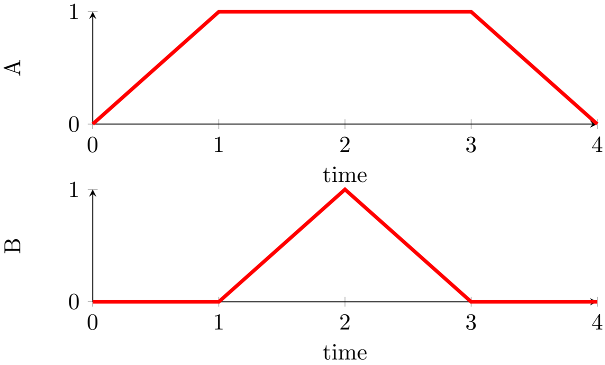


The control law

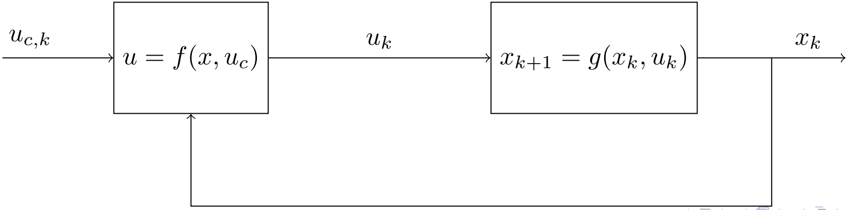


The lab assignment





controller = logic circuit   plant = pneumatic system



Implementing the sequence A+B+|B-A-, state variables

State variables

$$x = \begin{bmatrix} A_R & A_E & B_R & B_E & G_1 & G_2 \end{bmatrix}^T,$$

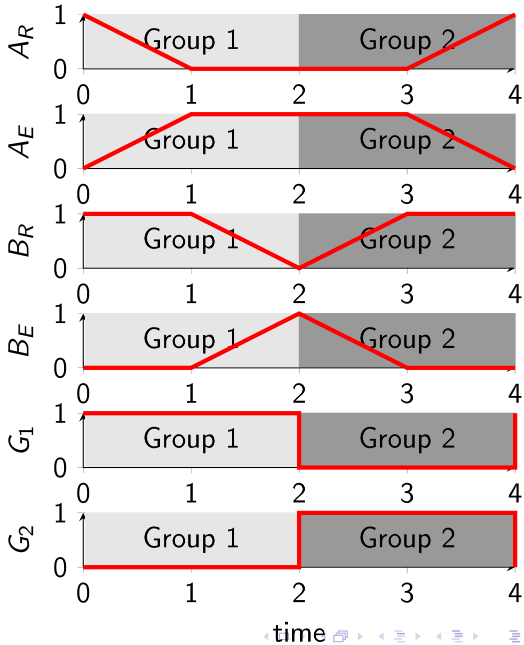
with

$$\{A_R, B_R\} = \begin{cases} 1 & \text{\{A,B\} retracted} \\ 0 & \text{\{A,B\} not retracted} \end{cases}$$

$$\{A_E, B_E\} = \begin{cases} 1 & \text{\{A,B\} extended} \\ 0 & \text{\{A,B\} not extended} \end{cases}$$

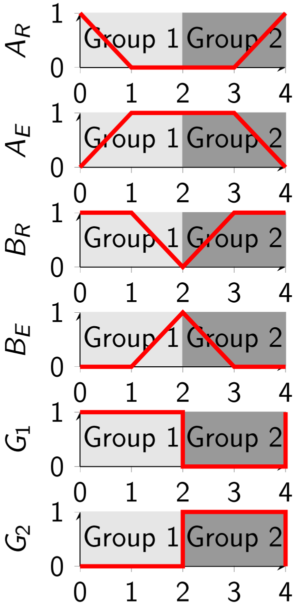
$$G_i = \begin{cases} 0 & \text{Group } i \text{ not active} \\ 1 & \text{Group } i \text{ active} \end{cases}$$

State transitions



Implementing the sequence A+B+|B-A-, control law

State transitions

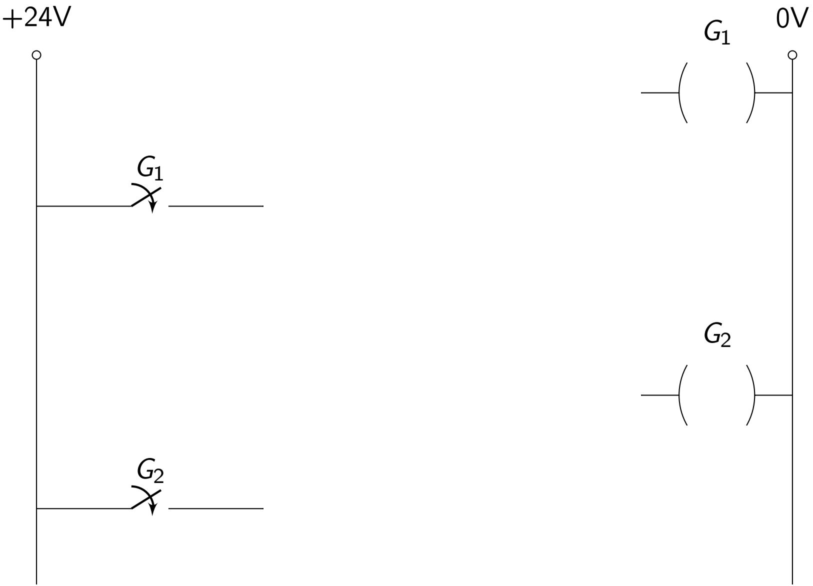


Control law

A_R	A_E	B_R	B_E	$G_1 1$	G_2	u_A+	u_A-	u_B+	u_B-
1	0	1	0	1	0				
0	1	1	0	1	0				
0	1	0	1	0	1				
0	1	1	0	0	1				

Implementing the control law

Implementing the group transitions



Implementing the proximity sensor circuit