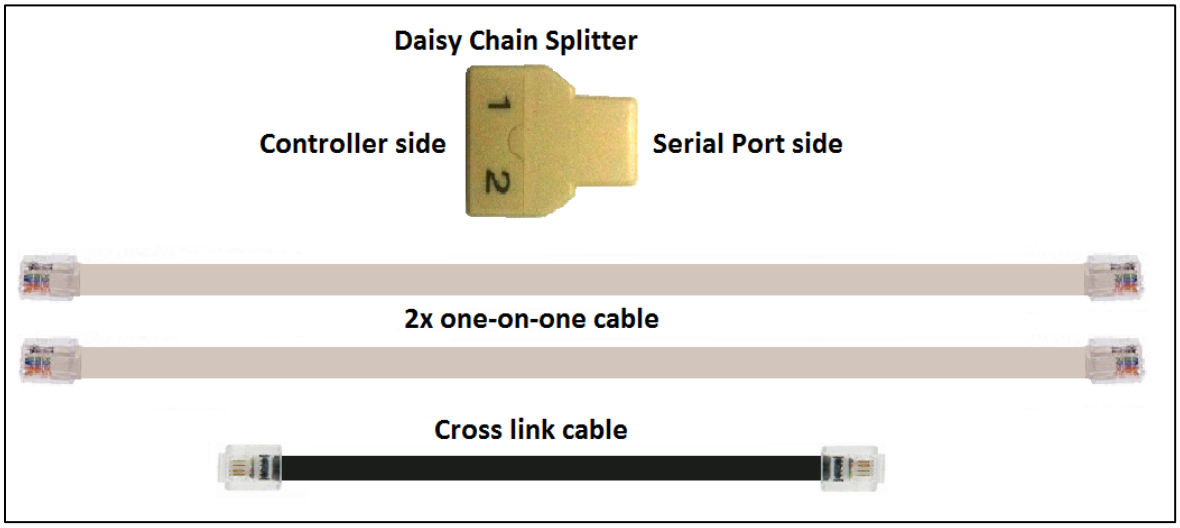
LCC–Daisy Chain Kit

LCC-Daisy Chain functionality

The Daisy Chain Kit contains:

1. Daisy Chain Splitter (containing 3 x RJ25 socket)
2. 2 x one-on-one cable (RJ25, Gray, 30 cm length , 6 wires)
3. 1 or more x Cross link cable (RJ14, Black cable, 20 cm length, 4 wires)



The number of cross link cables that you need depend on the amount of controllers in the chain. For 2 controllers you need only one cross link cable, for 3 controllers you need 2 cross link cables, for 4 controllers you need 3 cross link cables etc. The daisy chain kit has a part numbering system. The part numbering for a standard daisy chain kit is DCK-x1 . if you need more cross link cables the number behind the “x” indicates the amount of required cross link cables (DCK-x2 contains two cross link cables).

LCC-Daisy Chain functionality

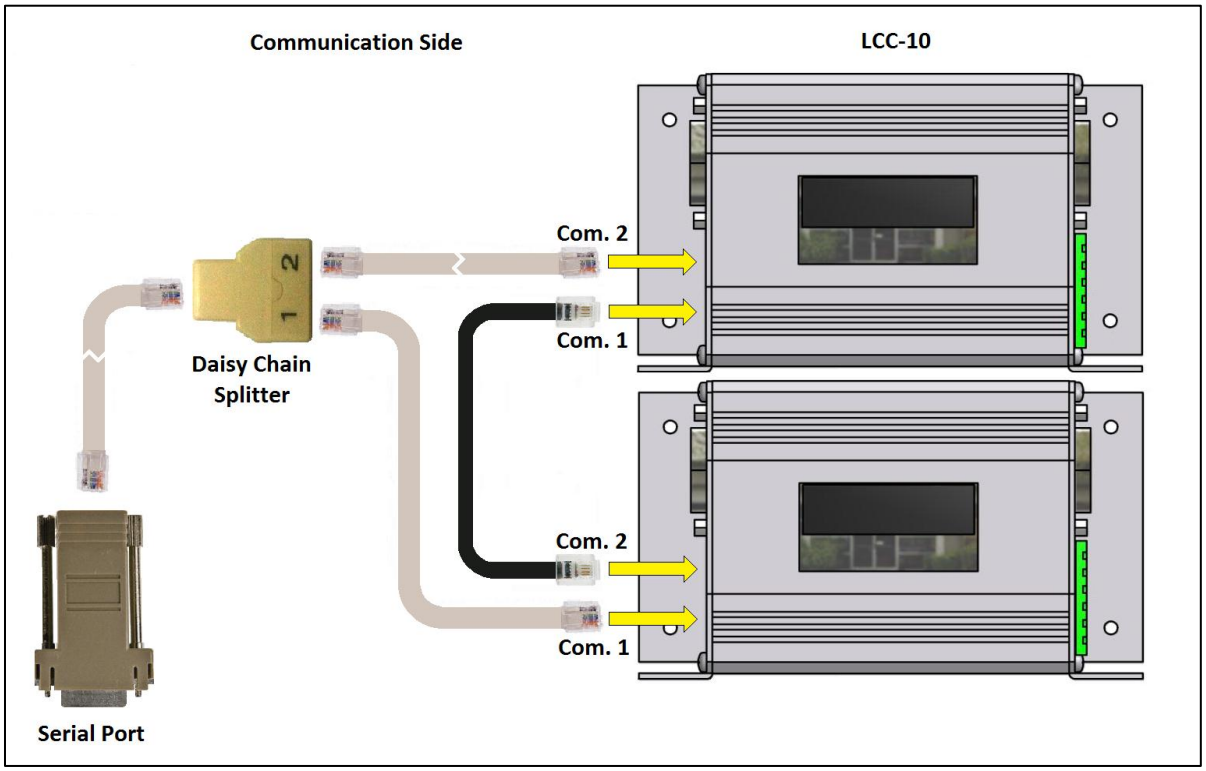
The daisy chain functionality provides the possibiltiy to use one serial port for multiple controllers. It does NOT enable direct communication between controllers. Since the LCC10 and LCC11 controller is only capable of acting as a slave the controller can not send commands to another controller. One can set up some form of communication between two controllers by using digital and/or analog handshaking. For the daisy chain to work each controller must have an unique node id and the daisy chain bit must be set for every controller in the chain. This bit is already set by standard config files.

Preparing LCC controllers for Daisy Chaining

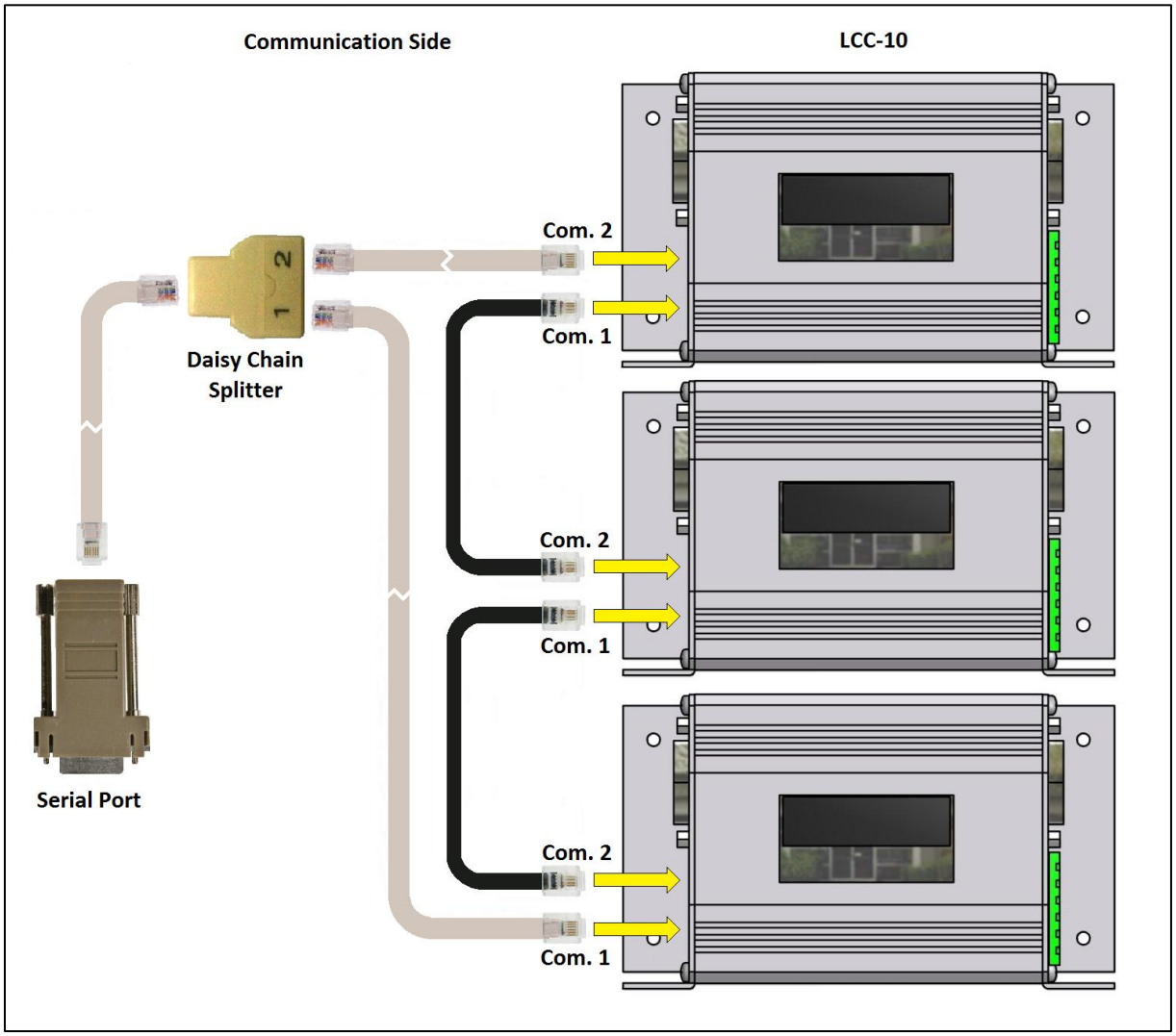
It is best to start to develop the application of each axis seperatly by connecting directly to the conttroller (no daisy chain used at that time). This means setting the node id (using MotionLab) or the axis, loading the configuration into the controller (using SMAC Control Center) and creating the program (using SMAC Control Center) . Then you can connect the daisy chain and by selecting the node id you can talk and receive messages from all different node id's on the chained network. Note that relaying one command to the next node can take about 12 ms when using firmware 1.0R. This means that if you would have 3 nodes (=3 controllers) and you would send out a message to all 3 simultaneously (node id 0 means all nodes) the first would respond 24 msec earlier than the third node in the network.

Connecting LCC controllers using Daisy Chaining

In the two images below the connections that need to be made are shown. For the connectivity to a serial port (of a PC or other device that is serving as a master controller) you also need a RS232 kit. Note that the gray one-on-one cables go to the first and last controller of the chain (channel 2 of the splitter must connect to com2 of the first controller and channel 1 of the splitter must connect to com1 of the last controller.



Daisy Chain with two controllers using DCK-x1

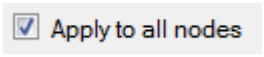


Daisy Chain with three controllers using DCK-x2

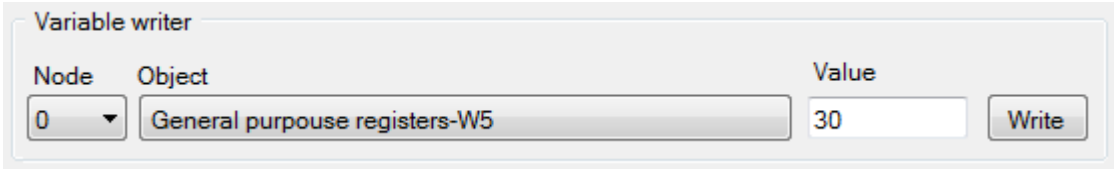
Programming using daisy chain

The biggest benefit from daisy chain is that programming multiple conrollers becomes easier. LCC control center can connect to one of the daisy chain controllers but a lot of functions can be executed on all controllers (=node nr 0) simultaneously. Functions that can be applied to all nodes simultaneously are:

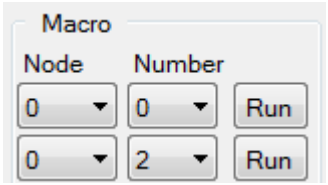
- Save in controller (in the build program tab) by checkmarking apply to all nodes



- Write an object to all controllers (in the run programs tab)



- Run the same macro on all controllers (in the run programs tab)



Note that all other node selectors are for one node only.

It is best to create one program file for all actuators and load it to all nodes simultaneously. If each node has a different program sequence, you can use the node number identifier in the program to select a sequence of macro's for each node. See the example below.

#	Line	Command	Parameter	Comment
0		MacroNumber	0	auto run macro
3	0-1	If	Variable,Var=Uart_configuration-Node_ID(0x012000),Equal,Const=1,Then_Macro_Jump=3,Else_Macro_Continue	Node nr = 1 => X boven
3	0-2	If	Variable,Var=Uart_configuration-Node_ID(0x012000),Equal,Const=2,Then_Macro_Jump=4,Else_Macro_Continue	Node nr = 2 => Y axis
3	0-3	If	Variable,Var=Uart_configuration-Node_ID(0x012000),Equal,Const=3,Then_Macro_Jump=5,Else_Macro_Continue	Node nr = 3 => Z axis
1	0-4	Macro	End_program	

In macro 0 there are 3 if statements. Each of them checks if a node number is in this controller. If so, it will execute the macro that belongs to that axis (node nr 1 is the x-axis and macro 3 is executed in that node).

Programming this way makes handling the program file easier.