HyValve

User Manual

Contact Details:

Tribotix Pty Ltd

87 Kirkdale Drive

Charlestown 2290

NSW Australia

P: +61 2 49578255

F: +61 2 49578244

W: www.tribotix.com

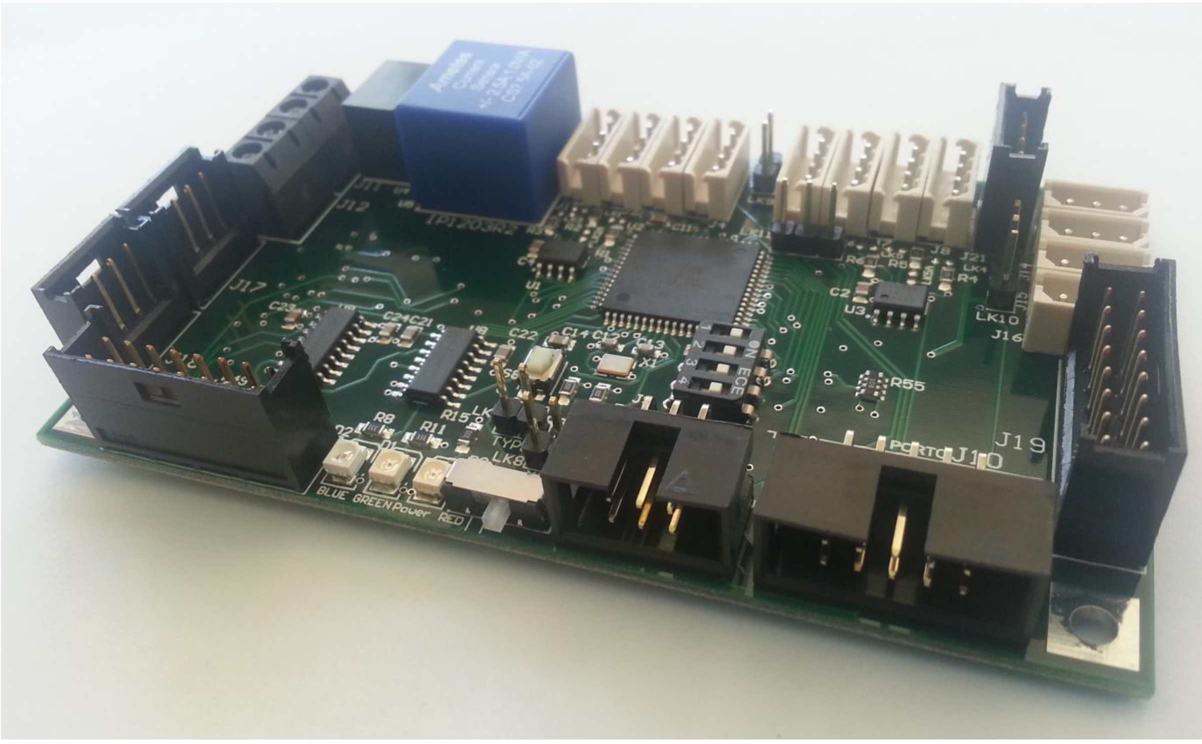
E: info@tribotix.com

Table of Contents

1. Overview

The HyValve Control board was designed to allow Robotis's Dynamixel modules to directly control Hydraulic Valves.

The HyValve control board processes signals from JoySticks, Encoders and Pushbuttons and sends positioning commands to Robotis's Dynamixel modules. Additionally open-collector digital outputs (capable of sourcing 15mA) are available to connect status LED's or connect to PLC's.



The HyValve control board is also capable of connecting to other HyValve boards via an RS485 network. This allows a great deal of flexibility in system design as data from all HyValve inputs can be shared with all HyValve's connected to the RS485 network.

The HyValve is based around the Atmel ATMega128 microcontroller and can easily be programmed with freeware software, such as Atmel's AVR Studio.

2. Hardware

The HyValve control board was developed specifically for controlling Hydraulic Valves via Dynamixel modules, a block diagram of the HyValve control board is shown below in figure 2.1.

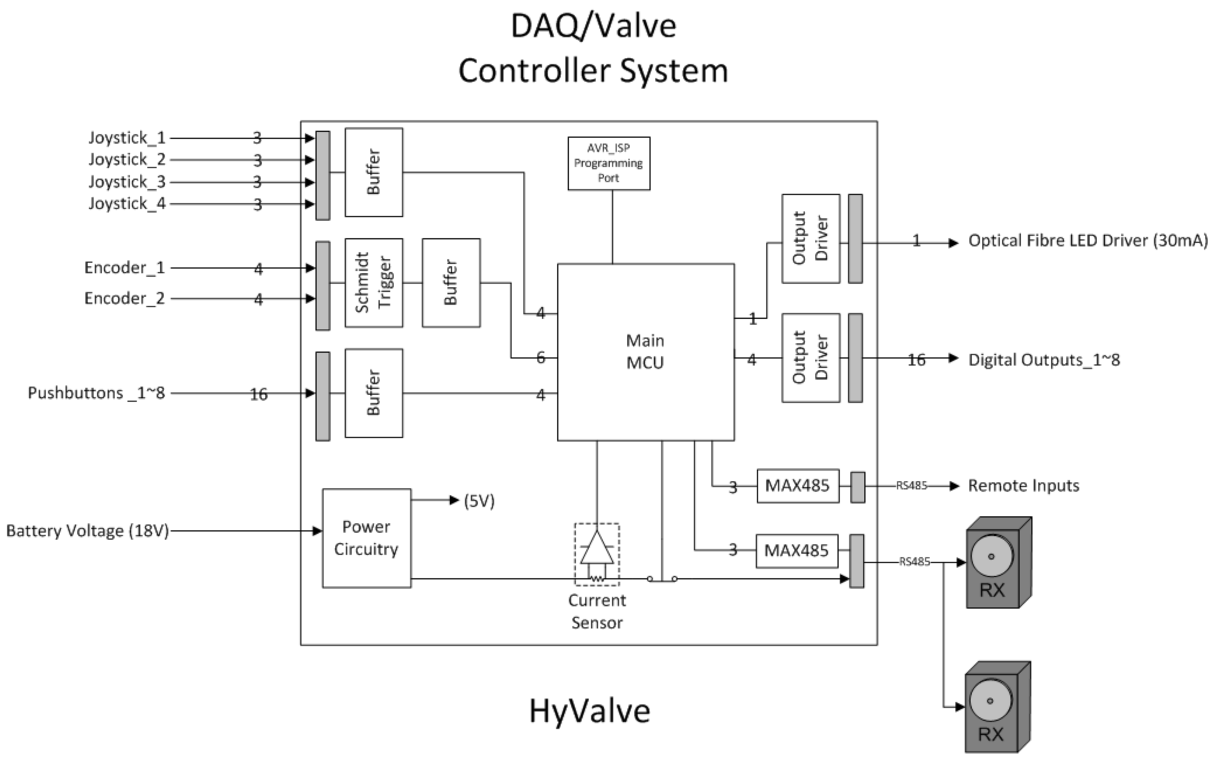


Figure 2.1 Block Diagram of single HyValve control board

The HyValve control board is capable of taking inputs from Joysticks, Encoders, and Push Buttons. The status of these inputs can then be used to control connected Dynamixel modules, which in turn are connected to Hydraulic Valves. Digital Outputs capable of driving LED's or interfacing to a PLC can also be used to show the status of the Dynamixels or the system as a whole.

The HyValve modules can be networked via an RS485 network, this creates a very flexible system where inputs connected to *any* HyValve control board can be used to control *any* Dynamixel connected to a HyValve control board. Figure 2.2 shows how multiple HyValve's may be networked.

The ability to connect multiple HyValve control boards via the RS485 network has the following advantages:

- No need for long cable runs, inputs and outputs can simply be connected to the closest HyValve and the data will be shared with connected HyValves.
- Reducing cable runs can minimise induced noise, particularly in analog/Joystick control signals.

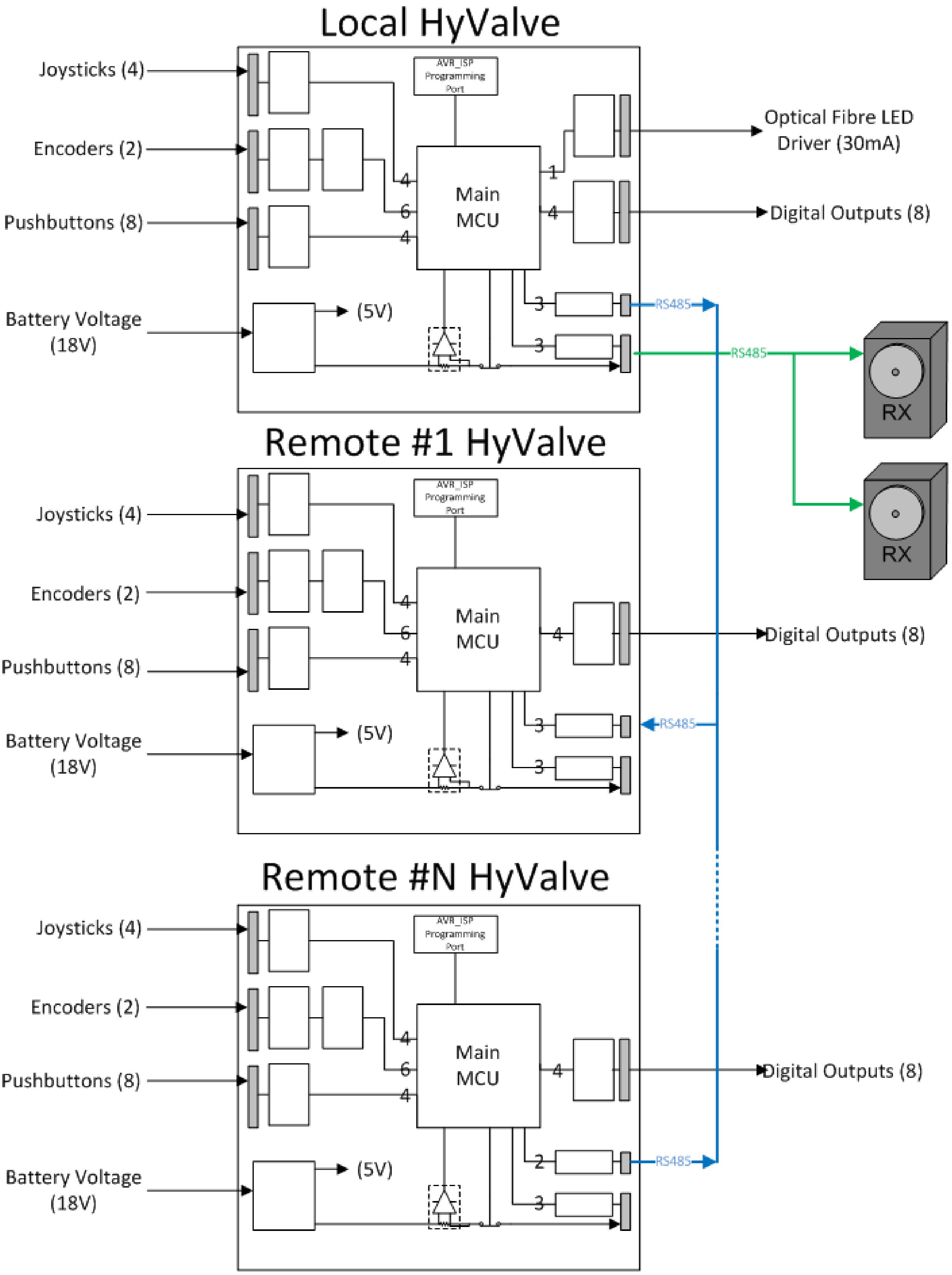


Figure 2.2 – Block Diagram of a networked HyValve configuration

2.1 Specifications

Power Supply:

V_{IN} – 12~18V DC

Inputs:

- RS485 communication port to communicate with remote HyValve's.
- 4 x 0~5V Analog inputs, allowing joystick control.
- 2 x 0~ V_{IN} Encoder inputs, allowing for:
  - switch input for pushbutton control (used to centre/reset the encoder reading), and
  - dual pulse train input (Pulse A & B) for rotational control.
- 8 x 0~ V_{IN} Digital inputs, allowing pushbutton control
  (programmable for CW/CCW movement and Pulse or Latching functionality)

Outputs:

- RS485 communication port to control a Dynamixel modules (RX, EX and MX modules supported),
- 8 x 15mA Digital Outputs, for LED signalling.
- 1 x 30mA Digital Outputs, for driving Optical Fibre LED.

Dimensions:

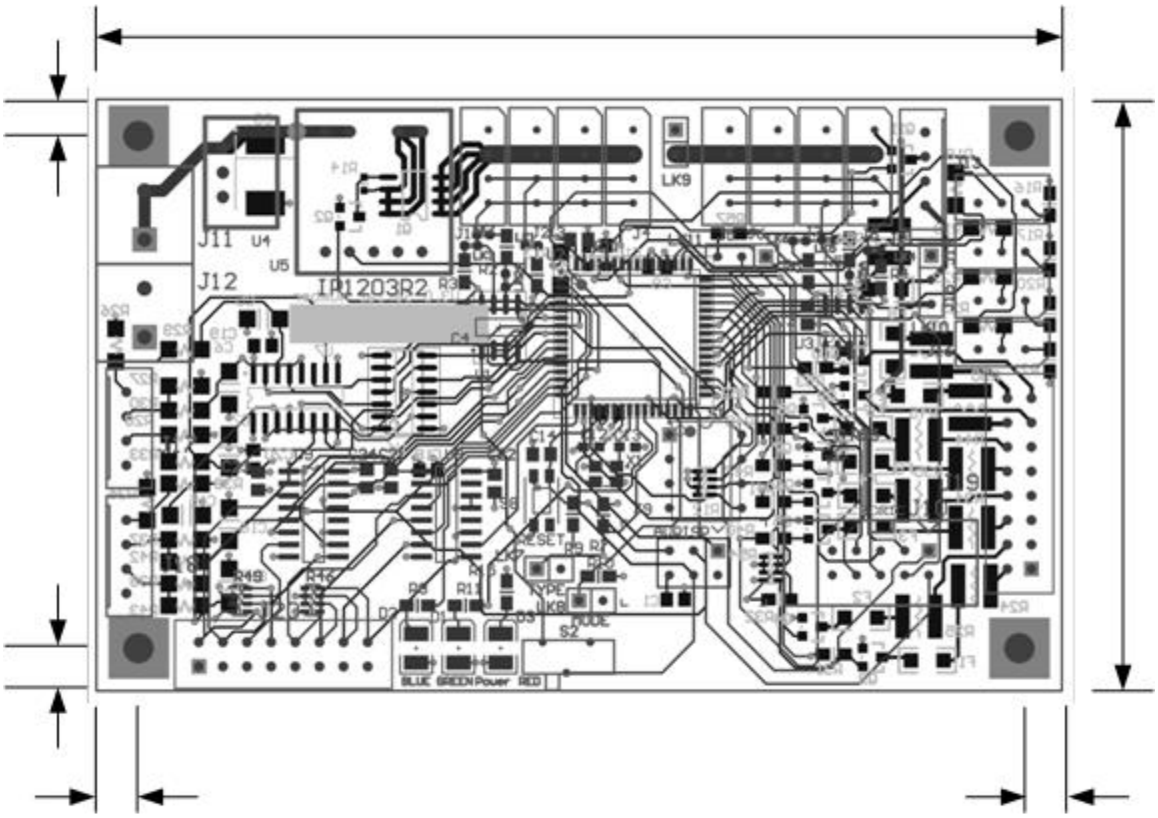


Figure 2.1.1 –Board Dimensions

2.2 HyValve Status LEDs

Three status LED's are provided on the HyValve control board, there location is shown in figure 2.2.1.

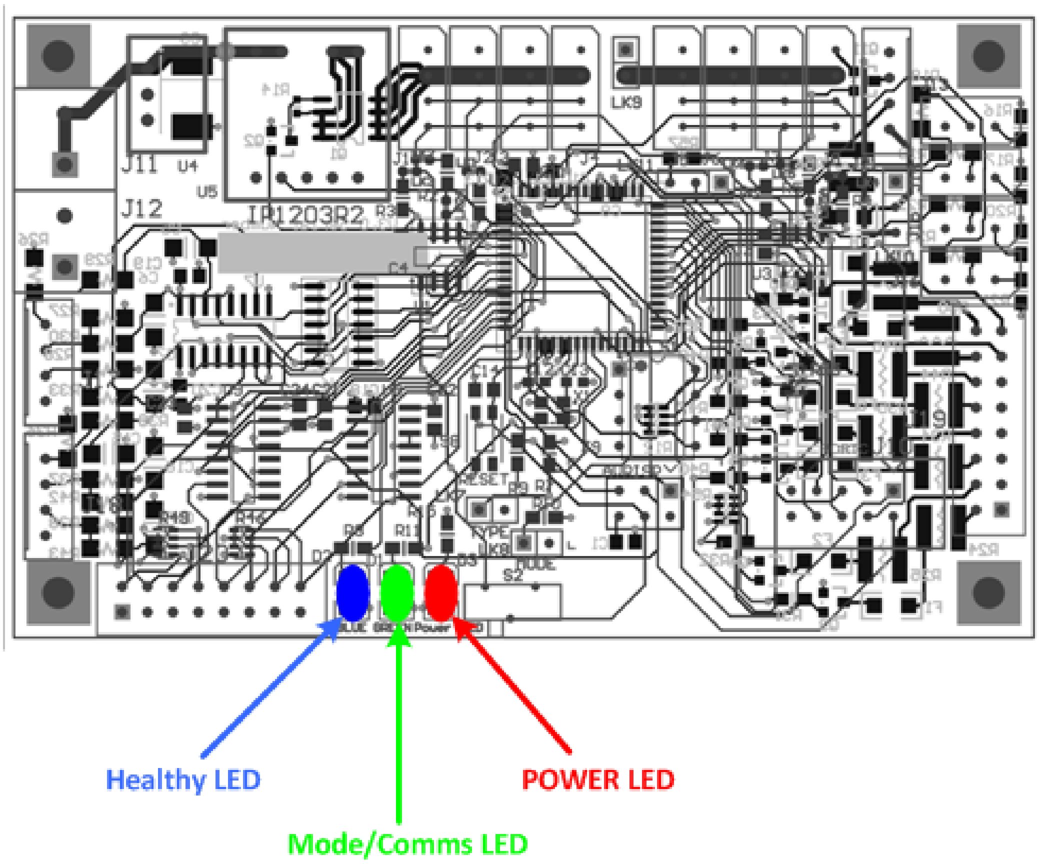


Figure 2.2.1 – HyValve Status LED's

POWER LED: RED

This LED is ON when power is applied to the HyValve control board.

Healthy LED: BLUE

This LED will FLASH each time the inputs are scanned and as such shows that the HyValve's software is executing as expected and the system is Healthy.

Mode/Comms LED: GREEN

The functionality of this LED depends on:

1. the MODE that the HyValve needs to operate in, and
2. whether the HyValve needs to be connected to the RS485 network.

The MODE jumper, discussed in a later section, is used to determine if the HyValve is a LOCAL or a REMOTE device. A LOCAL device has a Dynamixel connected whilst a REMOTE device is only used to gather the status of inputs and send this information to the LOCAL devices on the RS485 network.

<table>
<tr><th>Mode</th><th>Remote HyValve's connected</th><th>LED Functionality</th></tr>
<tr><td>LOCAL</td><td>NO</td><td>LED will be ON to signify it is in LOCAL mode</td></tr>
<tr><td>LOCAL</td><td>YES</td><td rowspan="2">LED will FLASH to signify that it has received a data packet from the RS485 network.</td></tr>
<tr><td>REMOTE</td><td>N/A</td></tr>
</table>

Table 2.2.1 – Mode/Comms LED Status

2.3 HyValve Power and IO connections

Figure 2.3.1 shows the physical layout of the HyValve control board, along with the locations of the external connections. The HyValve control board can be supplied with our without connectors, this is to account for applications, such as maritime installations, where better system reliability my be obtained by soldering wires straight to the pcb's.

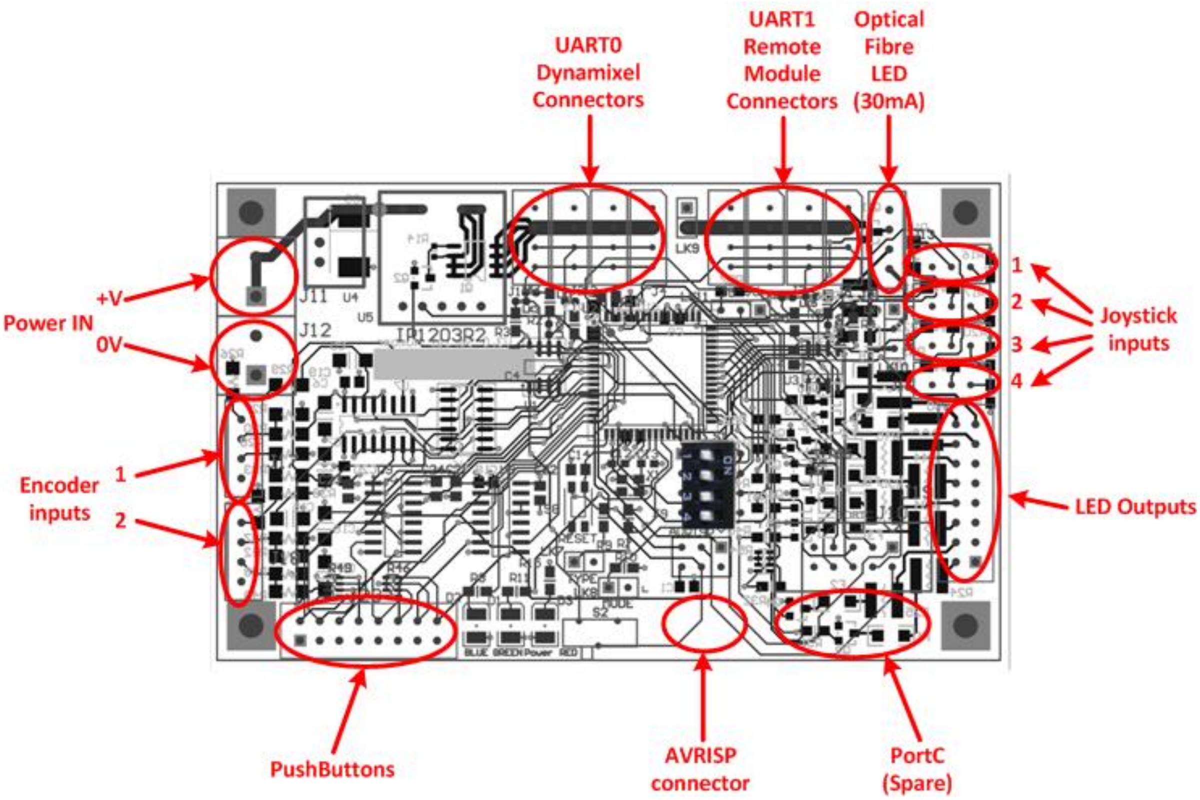


Figure 2.3.1 – HyValve Power and IO Connections

Inputs:

Power IN

Type: Screw Terminal
Manufacturer: RS
Part Number: 494-8978

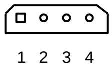
UART0 Dynamixel Connectors

Type: 4 way 2.5mm vertical header

Manufacturer: MOLEX

Part Number: 22035045

Pin Allocation:



Pin 1: 0V
Pin 2: +V
Pin 3: DA
Pin 4: DB

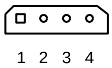
UART1 Remote Module Connectors

Type: 4 way 2.5mm vertical header

Manufacturer: MOLEX

Part Number: 22035045

Pin Allocation:



Pin 1: 0V
Pin 2: +V
Pin 3: DA
Pin 4: DB

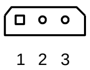
Joystick Inputs

Type: 3 way 2.5mm vertical header

Manufacturer: MOLEX

Part Number: 22-03-5035

Pin Allocation:



Pin 1: 5V
Pin 2: Analog In
Pin 3: 0V

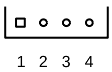
Encoder Inputs

Type: 4 x 1 way 2.54mm vertical shrouded header

Manufacturer: TE CONNECTIVITY / AMP

Part Number: 280371-2

Pin Allocation:



Pin 1: Switch
Pin 2: A
Pin 3: B
Pin 4: 0V

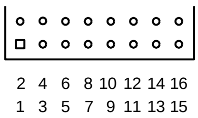
Pushbutton Inputs

Type: 8 x 2 way 2.54mm vertical shrouded header

Manufacturer: TE CONNECTIVITY / AMP

Part Number: 280385-2

Pin Allocation:



Pin 1:	0V
Pin 2:	PB1
Pin 3:	0V
Pin 4:	PB2
Pin 5:	0V
Pin 6:	PB3
Pin 7:	0V
Pin 8:	PB4
Pin 9:	0V
Pin 10:	PB5
Pin 11:	0V
Pin 12:	PB6
Pin 13:	0V
Pin 14:	PB7
Pin 15:	0V
Pin 16:	PB8

Outputs:

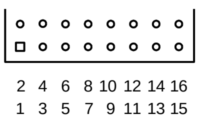
LED Outputs

Type: 8 x 2 way 2.54mm vertical shrouded header

Manufacturer: TE CONNECTIVITY / AMP

Part Number: 280385-2

Pin Allocation:



Pin 1:	0V
Pin 2:	LED1
Pin 3:	0V
Pin 4:	LED2
Pin 5:	0V
Pin 6:	LED3
Pin 7:	0V
Pin 8:	LED4
Pin 9:	0V
Pin 10:	LED5
Pin 11:	0V
Pin 12:	LED6
Pin 13:	LED 7 -ve
Pin 14:	LED7 +ve
Pin 15:	LED8 -ve
Pin 16:	LED8 +ve

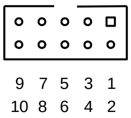
Port C (Spare)

Type: 5 x 2 way 2.54mm vertical IDC header

Manufacturer: TE CONNECTIVITY / AMP

Part Number: 1761681-3

Pin Allocation:



Pin 1: PC0
Pin 2: PC1
Pin 3: PC2
Pin 4: PC3
Pin 5: PC4
Pin 6: PC5
Pin 7: PC6
Pin 8: PC7
Pin 9: V_{cc}
Pin 10: 0V

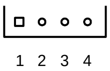
Optical Fibre LED 30mA Output

Type: 4 x 1 way 2.54mm vertical shrouded header

Manufacturer: TE CONNECTIVITY / AMP

Part Number: 280371-2

Pin Allocation:



Pin 1: V_{DD} or V_{CC} …. selectable via Jumper LK10
Pin 2: LED +ve
Pin 3: LED -ve
Pin 4: 0V

2.4 HyValve Jumper and DIP switch settings

Figure 2.4.1 shows the physical locations of jumpers and DIP Switches for the HyValve control board.

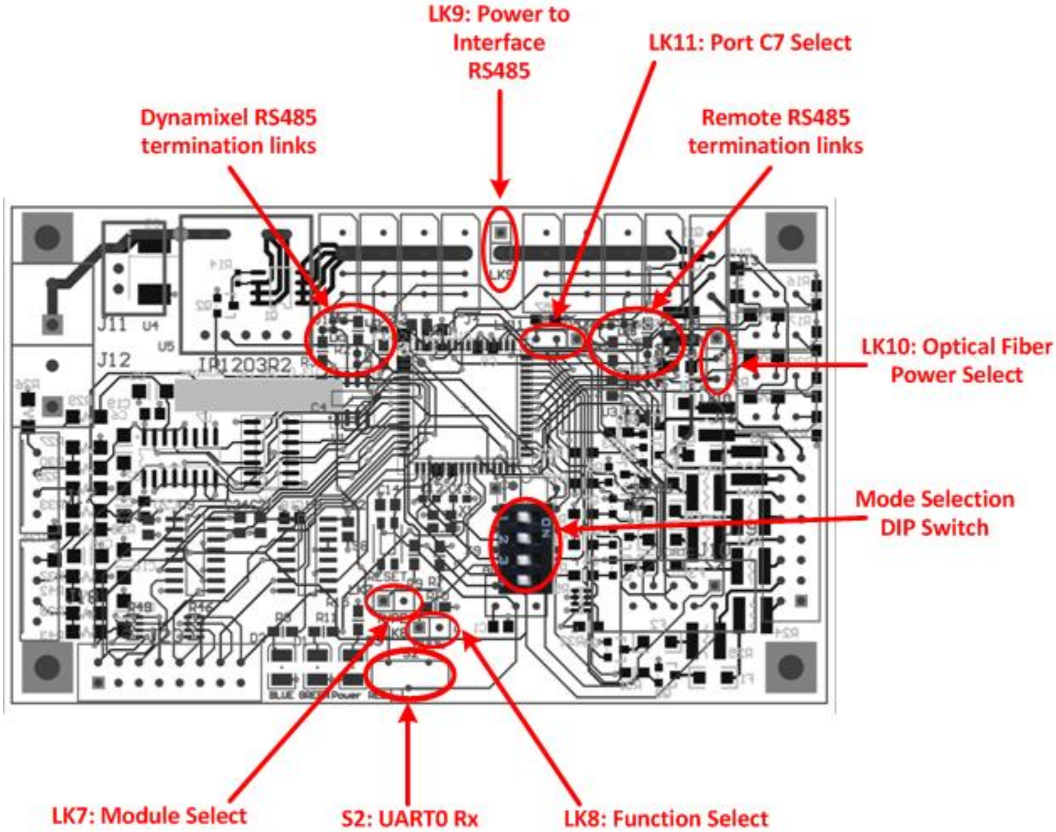


Figure 2.4.1 –Jumper and DIP Switch locations

Jumpers:

LK7: Module Select

LK7 – OFF, selects EX modules.
LK7 – ON, selects RX modules.

This jumper is used to determine which Dynamixel module is connected to the HyValve control board. The RX and EX modules have the following differences:

- RX modules use 10bit resolution for positioning the Dynamixel, i.e. 2^{10}=1024, over a 300^{o} control area, whilst
- EX modules use 12bit resolution for positioning the Dynamixel, i.e. 2^{12}=4096, over a 250^{o} control area.

Because of this difference the HyValve control board must now what type of Dynamixel is connected, so that the middle position, Clockwise (CW) and Counter Clockwise (CCW) positions can be defined.

LK8: Function Select

LK8 – OFF, selects Local mode.
LK8 – ON, selects Remote mode.

In situations where inputs are located a great distance from the Dynamixel module it is possible to gather this input information at a Remote Device and send this data via an RS485 communication network to a Local Device, which has control of the actual Dynamixel module.

HyValve control boards can operate in either Local or Remote modes, and the functionality is selected via LK2.

Regardless of a HyValve control board being in Local or Remote mode, the HyValve control board will always perform the following tasks:

- **Reads Inputs:** Joystick, Encoders and Pushbuttons,
- **Process Inputs:** determines required Dynamixel Position, and
- **Sets Outputs:** LEDs.

The difference in Local and Remote modes is that a:

- **Remote Device:** sends the status of it's IO and it's determined Dynamixel/Valve position to the Local Device, and
- **Local Device:** processes data from it's local inputs as well as any connected Remote Devices to determine the Dynamixel/Valve position.
  NB. Dynamixel modules are only ever connected to Local Devices.

NB. In a Local/Remote device RS485 network, the Local HyValve is the master of the communication network (i.e. it initiates all RS485 communications).

LK9: Power to Interface RS485

LK9 – OFF, Power disconnected from Remote RS485 cable.
LK9 – ON, Power connected from Remote RS485 cable.

There may be situations where it is easier to power a Remote Device via the +ve from the RS485 cable. If LK4 is ON then the +ve from V_{IN} will be connected to the +ve of the RS485 cable and allow the Remote Device to be powered by the RS485 cable.

LK10: Power select for Optical Fibre

LK11: Port C7 Select

Figure 2.4.2 shows the simple circuit used to generate the 30mA required by an external Optical Fibre LED. There are 2 selectable options:

1. LK10: Power supplied to J21 can either be V_{CC} (5V) or V_{DD} (16.5V),
2. LK11: MCU Output can be either connected to the base of the transistor Q11 (this is the default connection) or the signal can be passed to the Right Angled 10pin IDC Connector with all the other Port C signals.

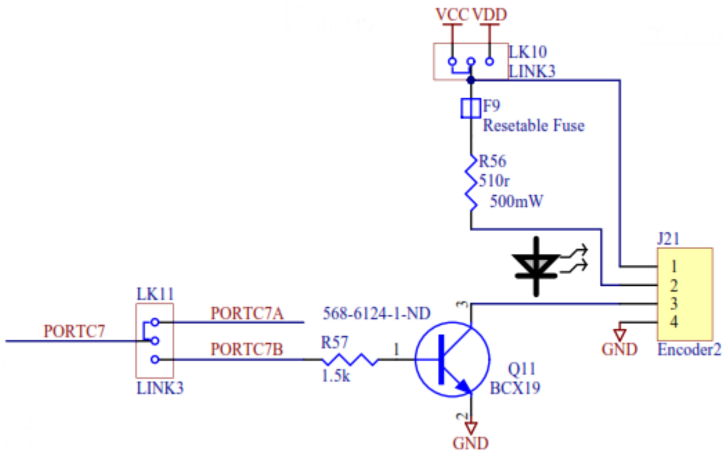


Figure 2.4.2 –Jumper setting for Optical Fibre 30mA Output

Switches:

S2: UART0 Rx

S2 – OFF, UART0 Rx disconnected from Dynamixel RS485 network.

S2 – ON, UART0 Rx connected from Dynamixel RS485 network.

When programming the ATMega128 microcontroller switch S2 must be in the OFF position. UART0 Rx is used as the RX signal from the Dynamixel module and is also used by the AVRISP for programming the microcontroller. Having additional circuitry connected to UART0 Rx can cause problems when attempting to program the microcontroller, so switch S2 has been included to disconnect UART0 Rx from all additional circuitry whilst programming.

DIP Switch:

A 4 way DIP switch is provided on the HyValve control board to allow the user to select from 16 different logic configurations. Table 2.1 shows the switch settings and the resultant logic configuration numbers.

	4	3	2	1
0	OFF	OFF	OFF	OFF
1	OFF	OFF	OFF	ON
2	OFF	OFF	ON	OFF
3	OFF	OFF	ON	ON
4	OFF	ON	OFF	OFF
5	OFF	ON	OFF	ON
6	OFF	ON	ON	OFF
7	OFF	ON	ON	ON
8	ON	OFF	OFF	OFF
9	ON	OFF	OFF	ON
10	ON	OFF	ON	OFF
11	ON	OFF	ON	ON
12	ON	ON	OFF	OFF
13	ON	ON	OFF	ON
14	ON	ON	ON	OFF
15	ON	ON	ON	ON

Table 2.1 DIP Switch Settings

The function of these DIP switch is determined by the programmer but common uses for the DIP Switches are:

1. Selection of predefines operating modes defining how the Dynamixel(s) react to the system inputs, and/or
2. Define Remote HyValve’s ID when used in a Networked configuration.

Solder Tabs:

When RS485 networks are connected over long distances it is important that the data signals are terminated properly and biased such that the data signals have the best chance of being recovered by the receiving node. RS485 networks are referred to as multi-drop networks, when the end nodes are configured correctly multiple nodes can then be connected to the data signals – these nodes don't need any additional configuration.

Figure 2.4.2 shows the correct way to configure an RS485 network, the important things to note are:

1. The DA DB network is terminated at the nodes that are physically at either end of the network, in Figure 2.4.2 this is nodes #1 and #N.
2. The terminating resistor for an RS485 network, R_T, is 120R.
3. Resistors R_A and R_B are equal valve (generally 10k) so that the data signals DA and DB can swing about $0.5V_{cc}$. This allows the data to be transmitted over a longer distance before the data is corrupted.

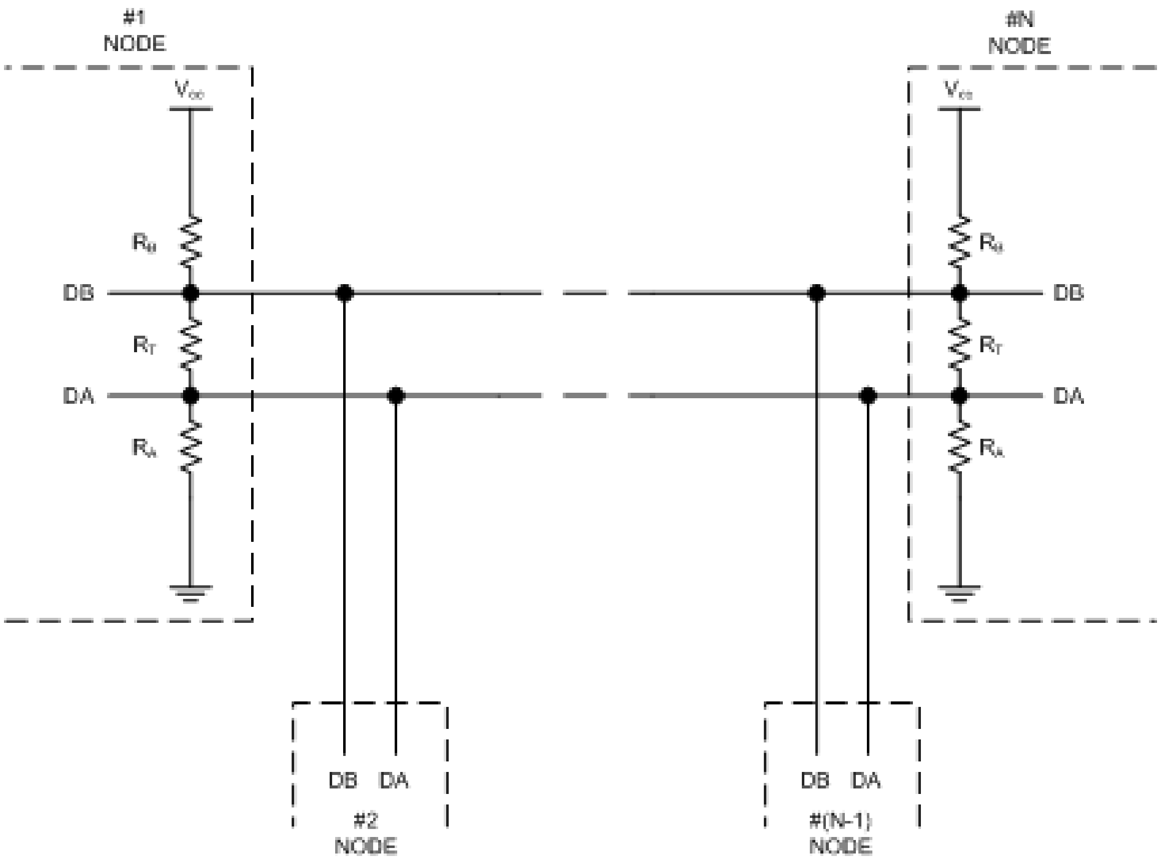
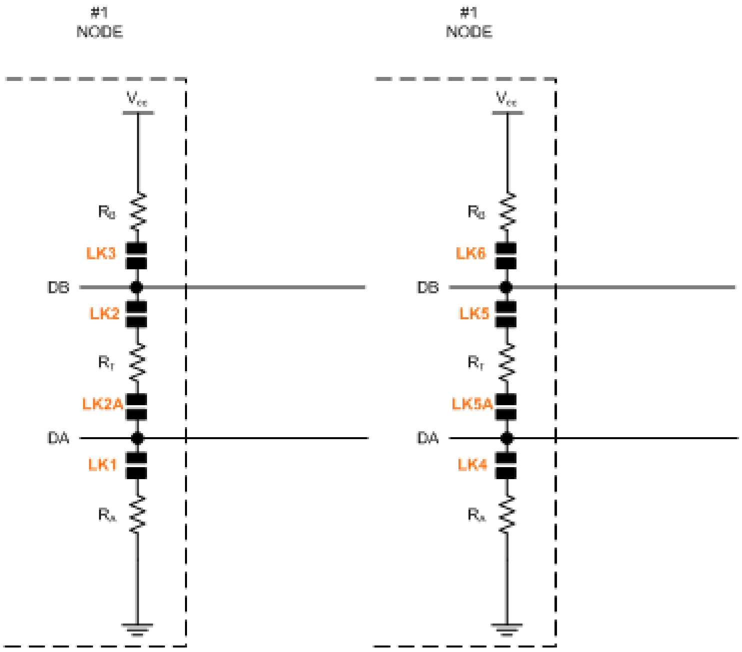


Figure 2.4.2 – RS485 Network

The HyValve control board has R_T, R_A and R_B provided on the circuit board, but as these resistors are only needed when the HyValve is one of the end nodes of an RS485 network, solder tabs are used to connect the resistors into the circuit. Figure 2.4.3 shows the circuits used on the HyValve's boards to configure the RS485 network termination and biasing.

If the HyValve is at the physical end of an RS485 network then:

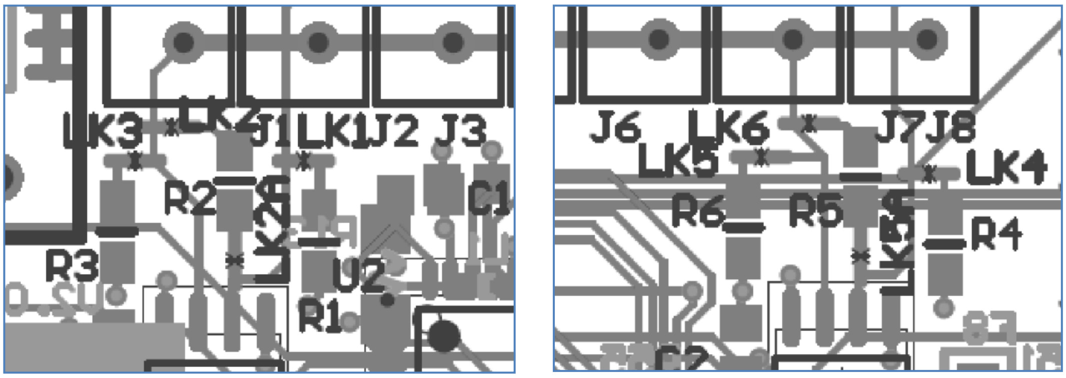
- UART0, Dynamixels: LK1, LK2, LK2A and LK3 must be connected, and
- UART1, Remote boards comms: LK4, LK5, LK5A and LK6 must be connected.



(a) **UART0** Dynamixels (b) **UART1** Remote boards comms

Figure 2.4.3 – HyValve RS485 configuration solder tabs

Figure 2.4.4 shows the physical locations of the solder tabs on the circuit board, as designated by the * symbol.



(a) **UART0** Dynamixels (b) **UART1** Remote boards comms

Figure 2.4.4 – Physical layout of HyValve RS485 configuration solder tabs

These solder tabs are:

- supplied OPEN CIRCUIT (as this is the most common configuration),
- if the HyValve is an END NODE then ALL 4 solder tabs must be connected, e.g. LK4, LK5, LK5A and LK6 must be connected.
- connection is made by placing a dob of solder across the solder tab.

NB. RS485 network termination and biasing is NOT needed when Dynamixels are connected to the HyValve control board, the solder tabs have only been provided on UART0 Dynamixels for consistency and possible future uses.

Example 1:

In this example **2** HyValve's are connected via the RS485 network, in this case **BOTH *Node #1* and *Node #2*** will need to have the solder tabs **LK4, LK5, LK5A and LK6 connected**.

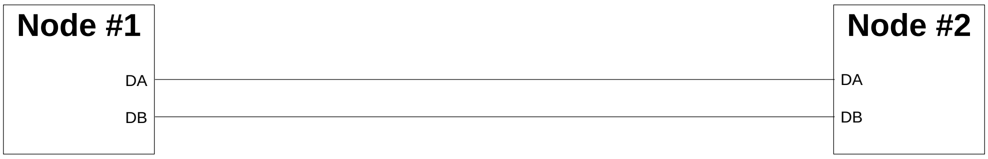


Figure 2.4.5 – Example 1: 2 HyValve's connected on RS485 Network

Example 2:

In this example **3** HyValve's are connected via the RS485 network, in this case **ONLY *Node #1* and *Node #3*** will need to have the solder tabs **LK4, LK5, LK5A and LK6 connected**. *Node #2* does not need any solder tabs connected as it does not need to provide a termination resistor or network biasing.

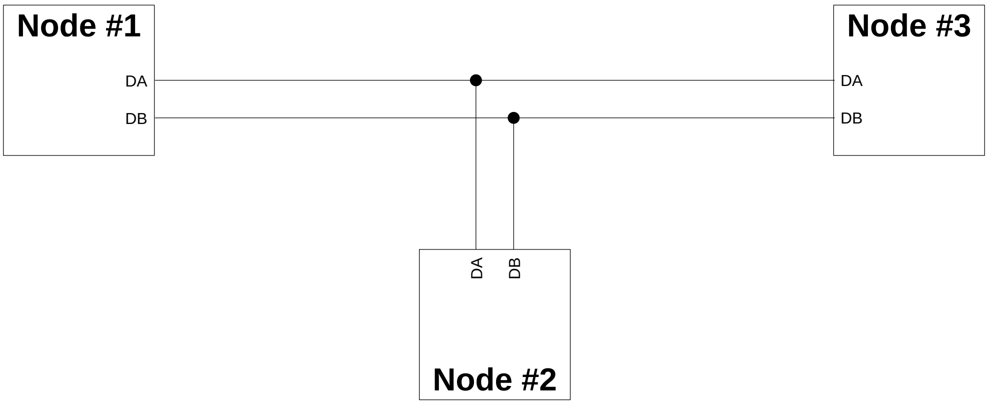
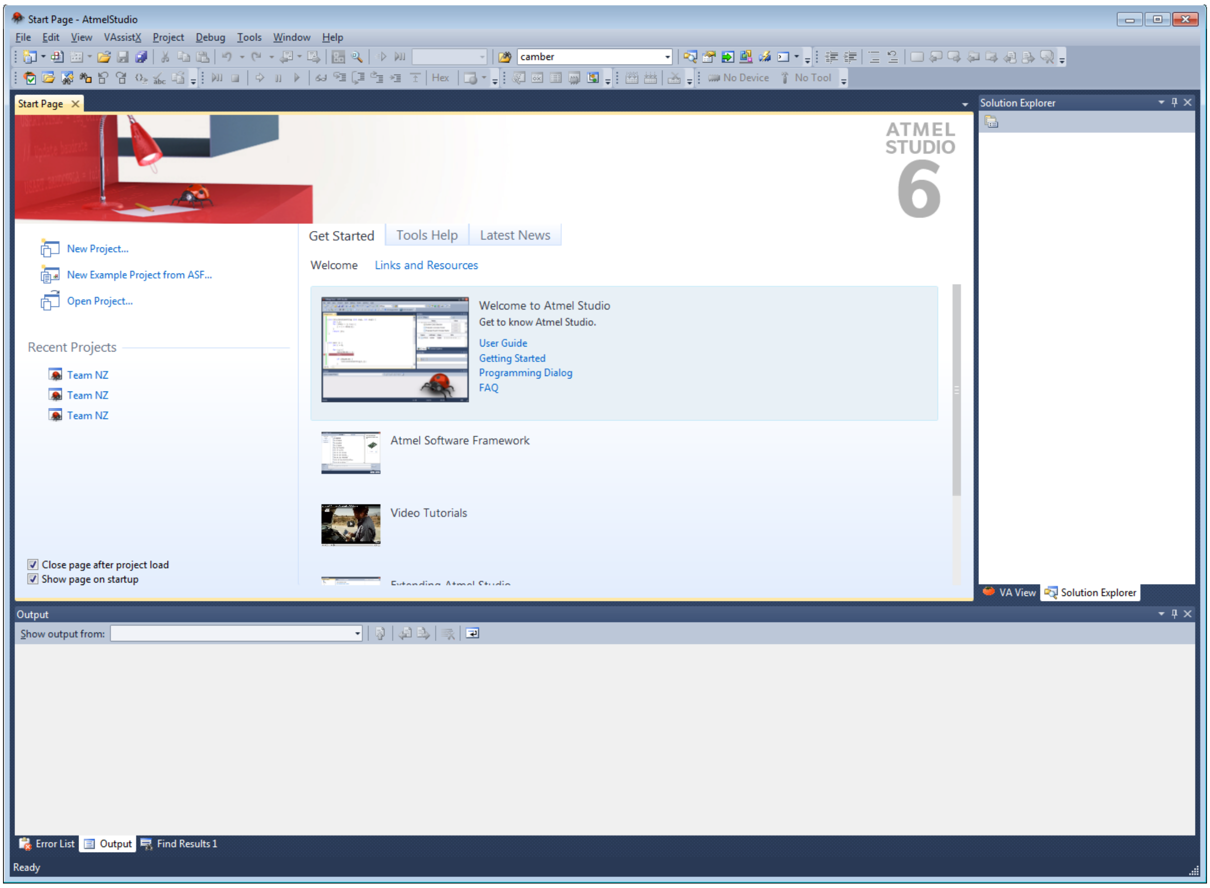


Figure 2.4.6 – Example 2: 3 HyValve's connected on RS485 Network

3. Software

The HyValve control board is based around the ATMega128 microcontroller from Atmel. Software can be written for the HyValve in C using Atmel's AVR Studio, software can then be downloaded to the microcontroller by Atmel's AVRISP.



NB. AVRStudio can be downloaded from the Atmel website.

3.1 Downloading code to the Microcontroller

3.1.1 Connecting AVRISP to HyValve

Figure 3.1.1.1 shows how to connect the AVRISP to the HyValve control board.

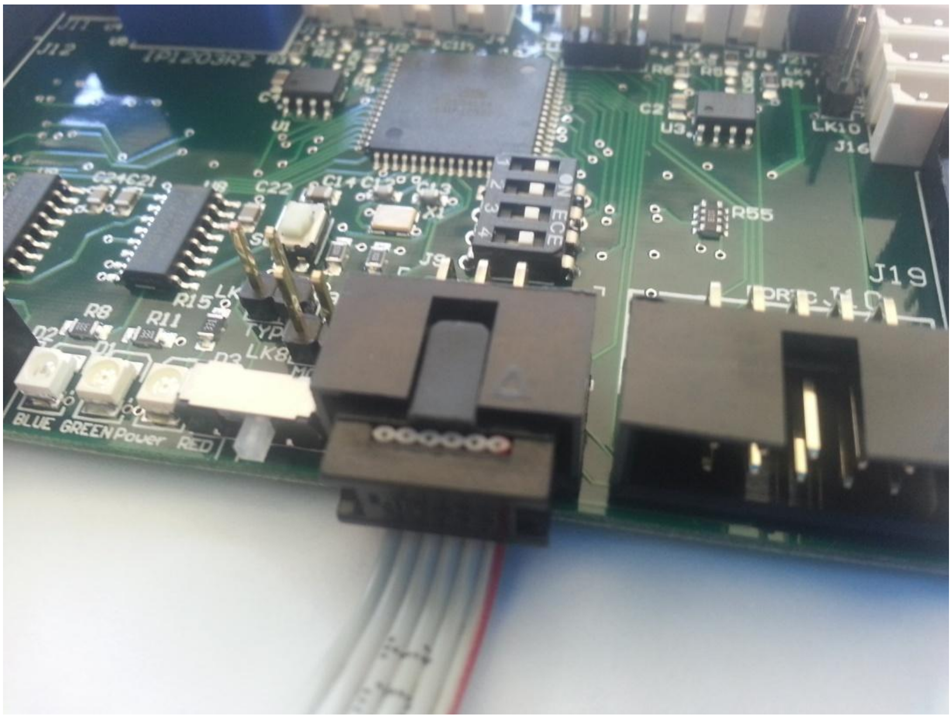
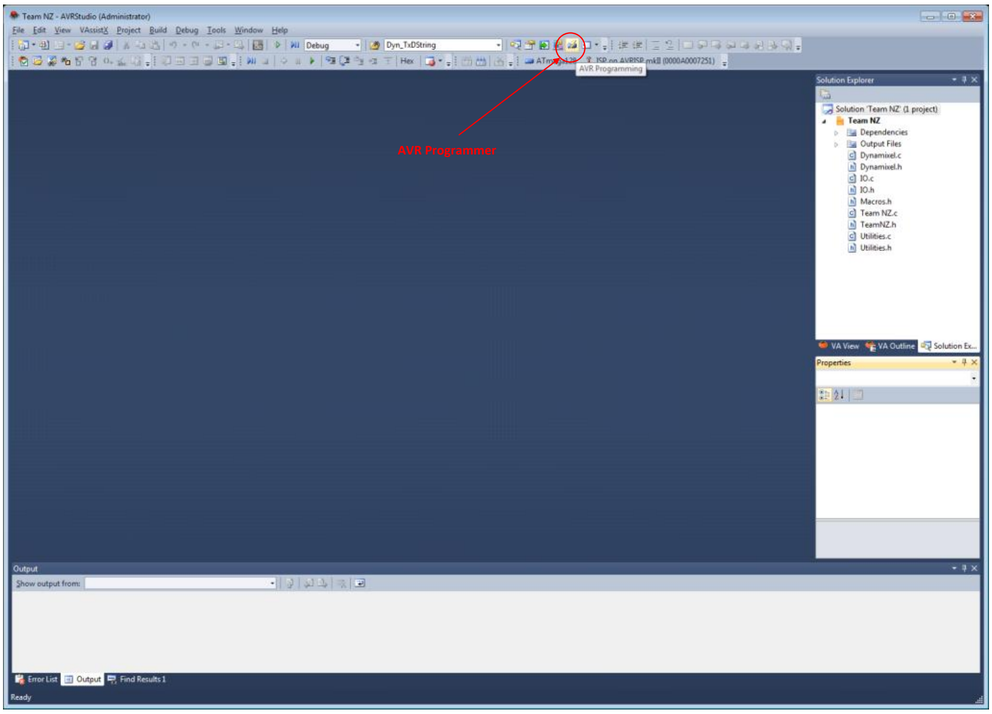


Figure 3.1.1.1 – Connecting the AVRISP to HyValve

To allow the AVRISP to talk to the ATMega128 the switch located adjacent to the AVRISP connector should be switched to the **RIGHT**, as shown in the figure above.

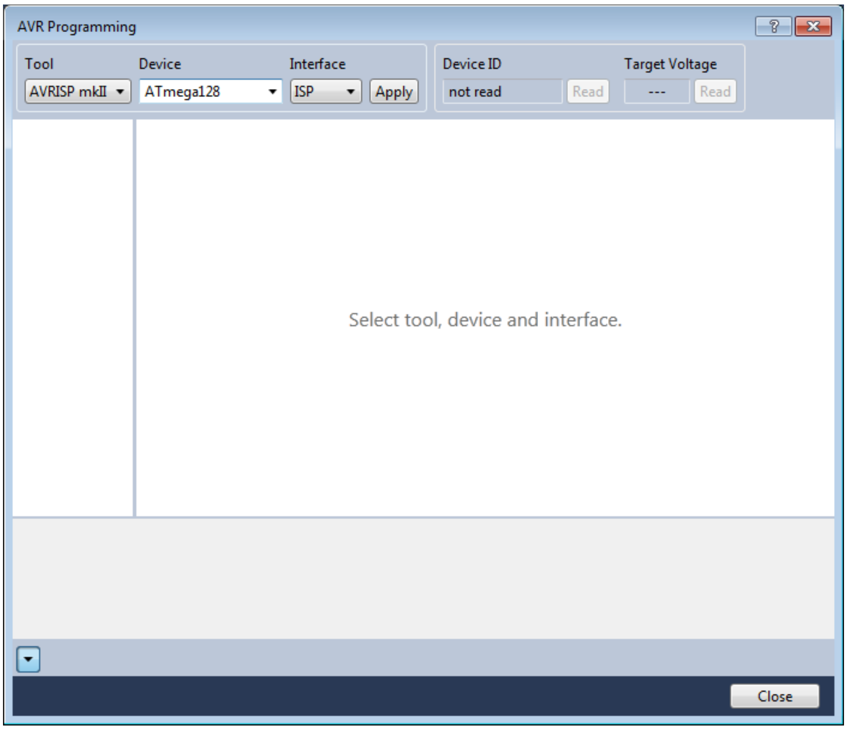
Once the AVRISP has been connected, the switch placed in the correct position and AVR Studio has been started the following procedure should be used to connect to the microcontroller:

1. Open the AVR Programmer with this icon as shown below.

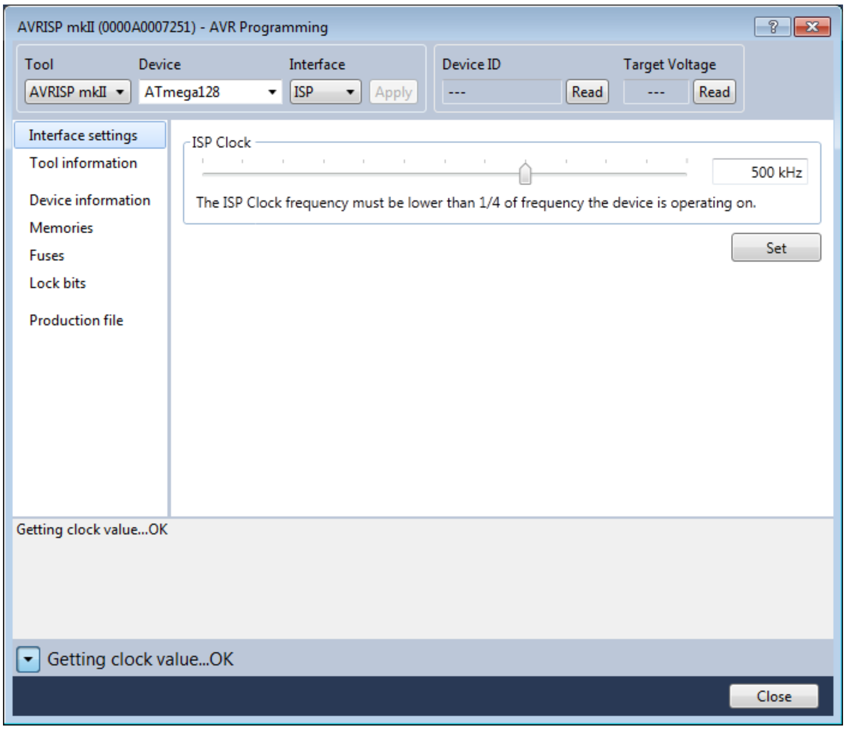


2. The Window shown below will then appear – make sure that:
   - the Tool selected is the AVRISP MkII,
   - the Device selected is the ATMega128.

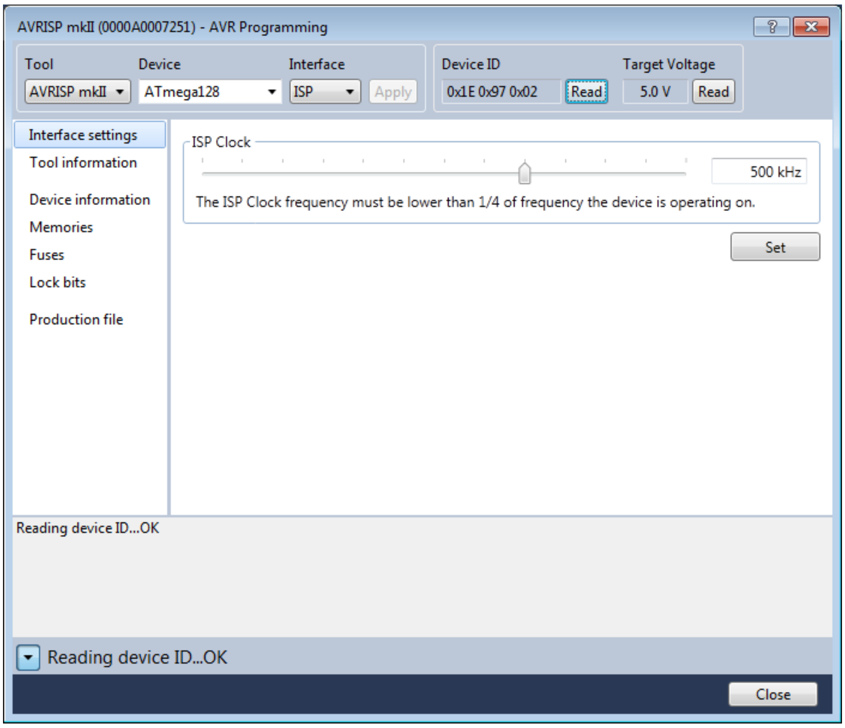
   Then hit *Apply* button.



3. When the AVRISP has connected to the ATMega128 the window will change as shown below. To verify the connection hit the *Device ID READ* button – this should show the 'signature' of the microcontroller.



4. The image below shows a correctly connected system where the *Device ID is 0x1E 0x97 0x02* – the AVRISP is now successfully talking to the microcontroller.



3.1.2 Microcontroller Settings

The ATMega has a series of internal Fuses and Lock bits that are used to configure the operation of the microcontroller. The settings required to operate with the HyValve control board are shown in figures 3.1.2.1 and 3.1.2.2.

NB. Before the HyValve control board is shipped these setting will be made within the ATMega128, this information is provided purely as a reference for the end user.

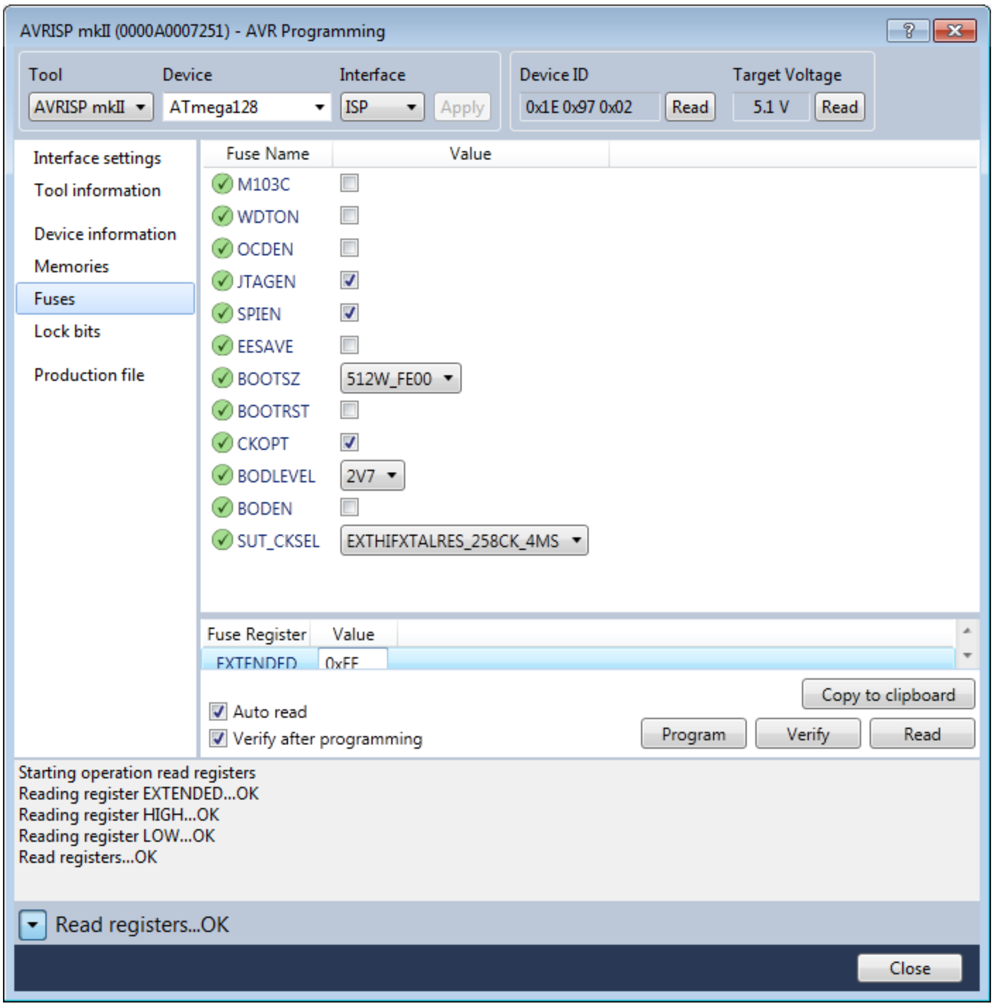


Figure 3.1.2.1 – ATMega128 Fuse settings

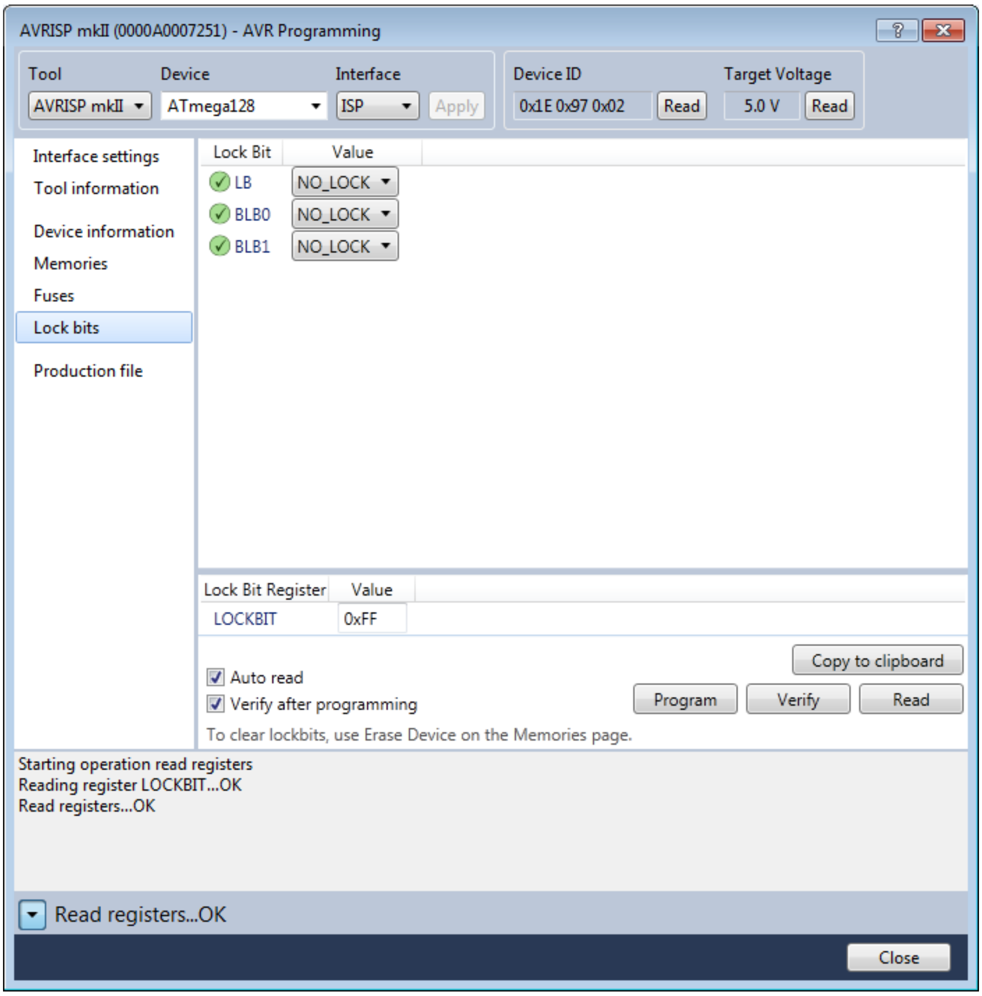


Figure 3.1.2.2 – ATMega128 Lock Bit settings

3.1.3 Microcontroller Programming

Once the AVRISP is communicating with the ATMega128 programming the microcontroller is quite simple. Figure 3.1.3.1 shows the window that is invoked when the *Device Programming* icon is pressed within AVR Studio.

Selecting the Memories option will display the window as shown below, simply select the hex file that has been created by AVR Studio in the *FLASH (128k)* drop down selection and press the *Program* button.

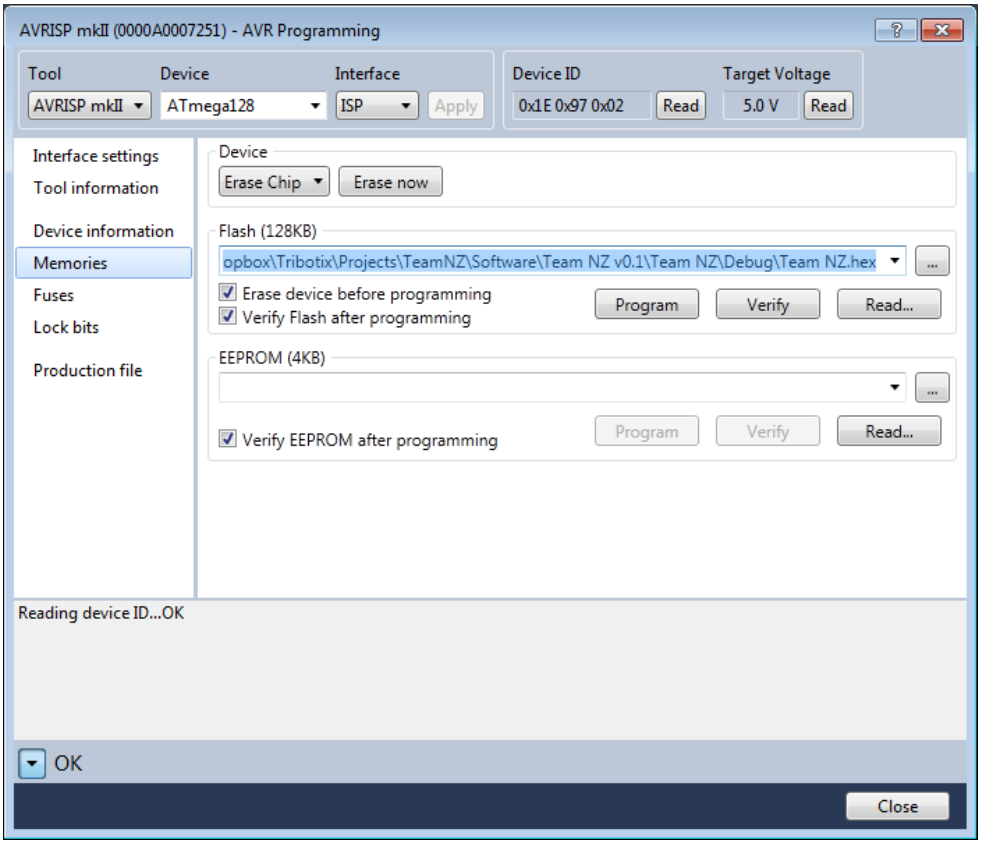


Figure 3.1.3.1 – ATMega128 Programming

As a rule-of-thumb we recommend that you select the *Verify Flash after programming* option to ensure that the microcontroller was programmed successfully.

Once successfully programmed the ATMega128 will self-reset and then start executing the new hex file automatically.

NB. The EEPROM (4KB) selection is not required unless your code has been configured to read data from the ATMega128's internal EEPROM.

3.2 Microcontroller IO Allocation

Table 3.2.1 shows the IO Allocation for the ATMega128, this information will be needed when writing software for the HyValve control board.

Pin			
51	PA0		Output_1
50	PA1		Output_2
49	PA2		Output_3
48	PA3		Output_4
47	PA4		Output_5
46	PA5		Output_6
45	PA6		Output_7
44	PA7		Output_8
10	PB0		PushButton_1
11	PB1		PushButton_2
12	PB2		PushButton_3
13	PB3		PushButton_4
14	PB4		PushButton_5
15	PB5		PushButton_6
16	PB6		PushButton_7
17	PB7		PushButton_8
35	PC0		Spare_1
36	PC1		Spare_2
37	PC2		Spare_3
38	PC3		Spare_4
39	PC4		Spare_5
40	PC5		Spare_6
41	PC6		Spare_7
42	PC7		Spare_8
25	PD0	SCL	Mode/Comms LED
26	PD1	SDA	Dynamixel Type
27	PD2	RXD1	Interface_Rx
28	PD3	TXD1	Interface_Tx
29	PD4		Interface_En
30	PD5		Mode
31	PD6		Encoder1_S
32	PD7		Encoder2_S
2	PE0	RXD0	Dynamixel_Rx
3	PE1	TXD0	Dynamixel_Tx
4	PE2		Dynamixel_En
5	PE3		Healthy_LED
6	PE4	INT4	Encoder1_A
7	PE5	INT5	Encoder1_B
8	PE6	INT6	Encoder2_A
9	PE7	INT7	Encoder3_B
61	PF0	ADC0	JoyStick1
60	PF1	ADC1	JoyStick2
59	PF2	ADC2	JoyStick3
58	PF3	ADC3	JoyStick4
57	PF4	ADC4	N/C
56	PF5	ADC5	N/C
55	PF6	ADC6	N/C
54	PF7	ADC7	N/C
33	PG0		VDynON
34	PG1		DIPSwitch_1
43	PG2		DIPSwitch_2
18	PG3		DIPSwitch_3
19	PG4		DIPSwitch_4

Table 3.2.1 – IO Allocation for ATMega128

3.3 Recommended Control Algorithm

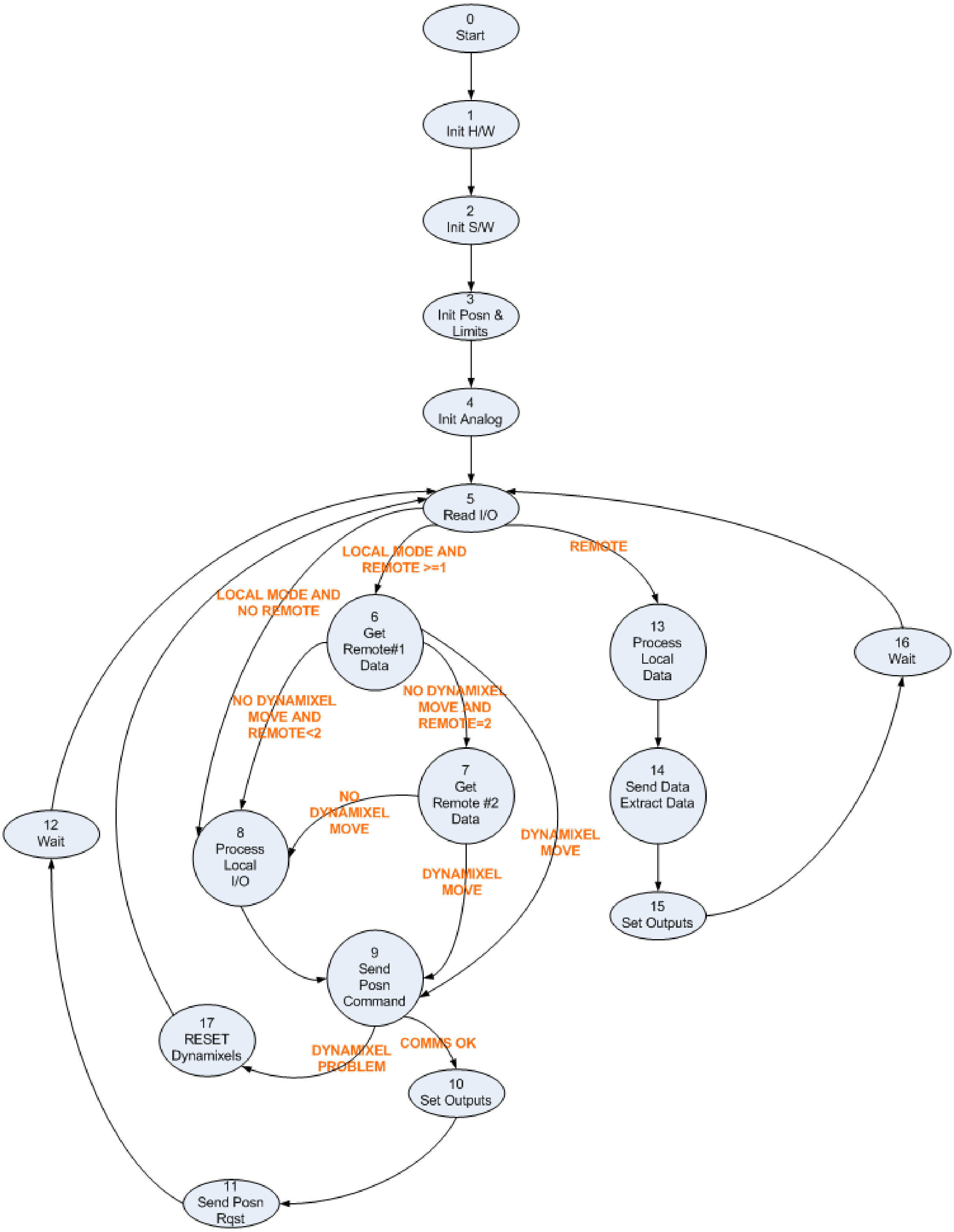


Figure 3.3.1 – Recommended Control Algotithm for System with 1 Local and 2 Remote HyValves