Real Time Laboratory Mk II
Technical Guide

Adrian P. Robson and William D. Henderson

Version 4 – 5 March 2011

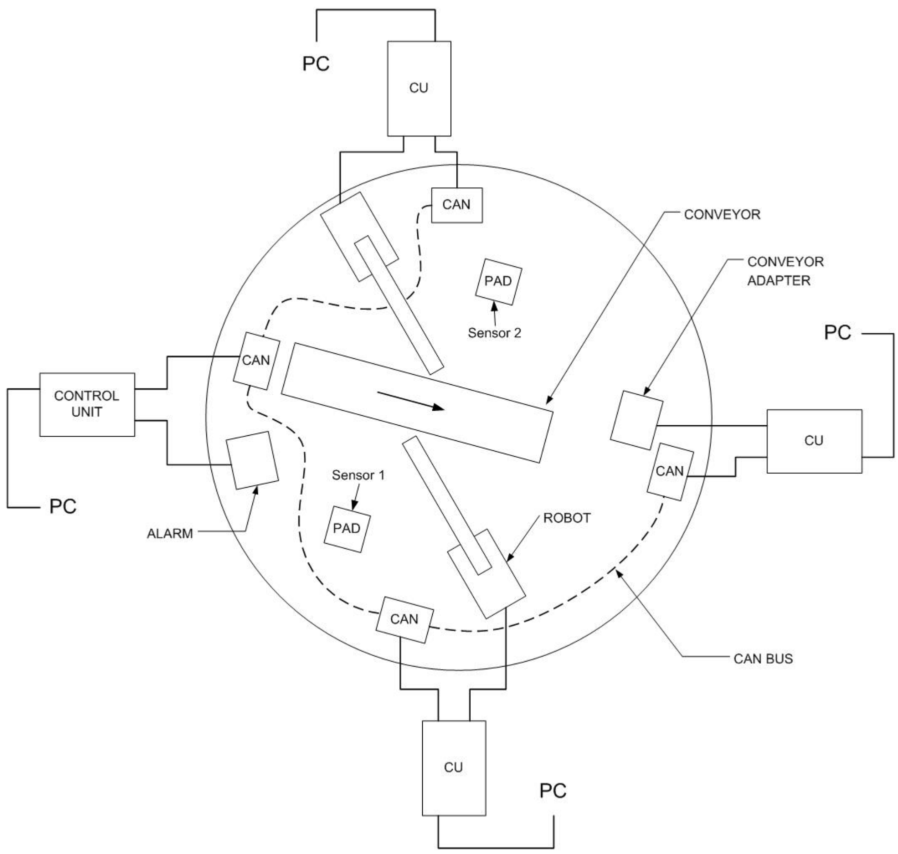
Contents

1 Introduction

The cell consists of two robots, a conveyor, an alarm/sensor module; and a CAN bus. The cell's devices are managed by control units that communicate with each other via the CAN bus. A particular ARM based control unit is described here, but any interface compatible technology could be used with the cell.

A case study will give much more detail, but normally the cell simulates a manufacturing process by moving small 'work pieces' from an input pad to an output pad. This is done by the cooperative operation of the robots and the conveyor. Various sensors provide information about the location of work pieces in the cell.

Physically the cell's components and the control units are arranged approximately as follows:



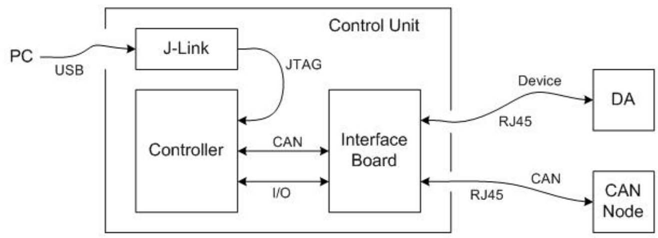
Each PC is connected to the control unit on its left. Each control unit connects to a PC, its device and a CAN bus (see §2.1). The control unit must be configured to work with its device by stetting some DIP switches (see §2.2). The CAN bus can be configured as one or two networks (see §5), and the control units must all be on the same network.

2 ARM Control Units

These are built with ARM based Olimex LPC-2378-STK boards. The boards are packaged with an interface module, which provides access to CAN bus and the controlled device, and a JTAG programmer.

2.1 Hardware Set-up

Control units are connected to a PC and the cell as follows:



There are three connections required:

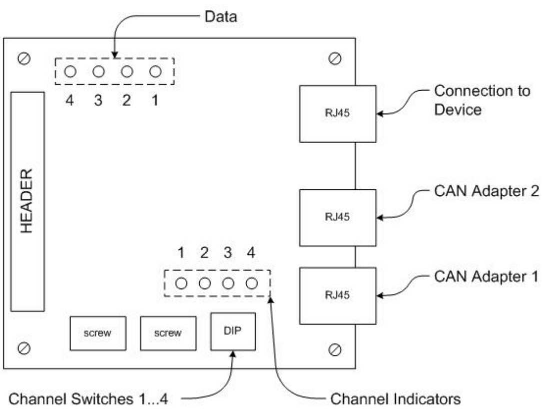
Programming
: Connection to a USB port on a PC with suitable development software. This link also provides the unit's power.

Device control
: RJ45 connection to a device adapter on the cell.

Communications
: RJ45 connection to a CAN node on the cell.

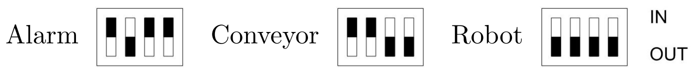
2.2 Interface Configuration

The controller interface has four signal channels that can be configured by switches to be input or output. An indicator LED is lit if a channel is in output mode. *Great care should be taken not to connect a channel in output mode directly to ground or the power line.*



The switches have to be configured for the particular device being managed. For the cell, this is done by setting the switches as follows:

Switch	Alarm/Sensor	Conveyor	Robot
1	in	in	out
2	out	in	out
3	in	out	out
4	in	out	out



where black represents a switch head.

> Great care must be taken to ensure that the switches are correct. A mistake can result in permanent damage to the interface and device adaptor hardware.

2.3 On Board LEDs

The controller interface has four data LEDs that can be managed by the controller. If a suitable software interface is not available (see §4), then appendix A should be consulted for the ARM I/O port allocations.

2.4 Robot Programming

Loading a program into an ARM controller that is managing a robot can cause the robot's servos to jitter. The following method should be used to avoid this:

- Remove the cable from the 'interface' socket on the controller interface board.
- The robot's servos will relax, and the robot can be moved by hand. Position it so that the arm joints are near mid-position.
- Load the new software into the controller, and start it.
- Reconnect the interface cable. The servos will energise and the robot will take up its initial programmed position, which should be close to the prepared position.

3 Power Management

If a cell is powered up, all of its control units should also be powered. A cell should not be powered when its associated PCs are switched off. When not in use the cell and its control units should be powered down as follows:

1. Switch off master power on cell.
2. Remove PC USB link from all control units.

Powering up the system is the inverse of this procedure.

4 Software Support

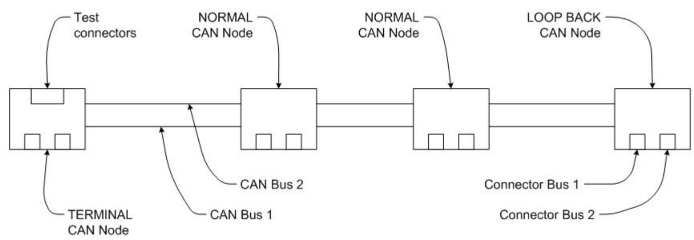
Software libraries and programming interfaces in a suitable programming language that support the controller interface and device adapter should be available. However, this guide does not document such facilities.

If suitable software interfaces are not available, the device adapter and the interface board's on-board LEDs must be managed directly by programming the ARM controller's I/O ports. Appendix A describes the mapping between ARM controller I/O ports and the controller interface; and §6 gives information about the device I/O channels.

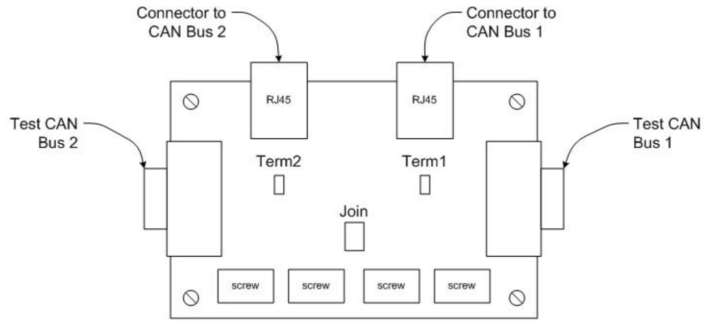
5 CAN Bus

The CAN bus on the production cell has connection points called *nodes*. There are four nodes in each cell each with two bus connectors, which are RJ45 sockets. The installation can be configured as a single bus with two connectors at each node, or as two independent buses each with a single connection at each node.

There are three types of node: terminal, loopback and normal. They all use the same PCB and are configured with jumper links to behave appropriately. The normal and terminal nodes must not be changed once they are installed in the cell, but the loopback node can be reconfigured to select single of double bus topologies. The terminal node has test sockets fitted so that an analyser can be connected to either of the buses.



The jumper settings for the different modes are as follows:



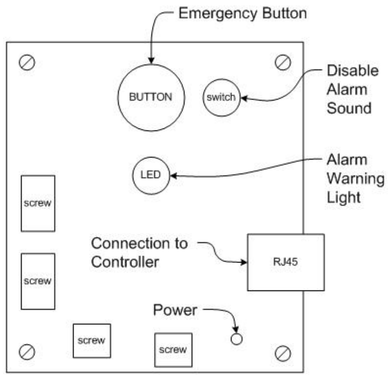
		Jumpers		
Node	Mode	Term1	Term2	Join
Loopback	Single	open	open	closed
Loopback	Double	closed	closed	open
Normal		open	open	open
Terminal		closed	closed	open

6 Device Control

6.1 Alarm/Sensor

6.1.1 Indicators, Switches and Connectors

I/O	Meaning
Button	Press⇒Emergency
Switch	On⇒Make sound on alarm Off⇒No sound on alarm
Light	On⇒Emergency or warning Off⇒No problem
Power	On⇒Board has power



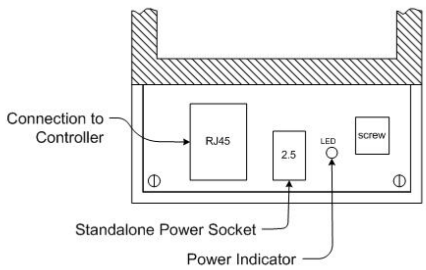
6.1.2 Channel Input and Output

Channel	Role	Mode	Logic
1	Button	in	0 ⇒ pressed
2	Alarm	out	1 ⇒ on
3	Sensor 2	in	0 ⇒ present
4	Sensor 1	in	0 ⇒ present

6.2 Robot

6.2.1 Indicators and Connectors

Indicator	Meaning
Power	On ⇒ Board has power



6.2.2 Channel Input and Output

Channel	Role	Mode	Logic
1	Waist	out	pwm
2	Elbow	out	pwm
3	Wrist	out	pwm
4	Hand	out	pwm

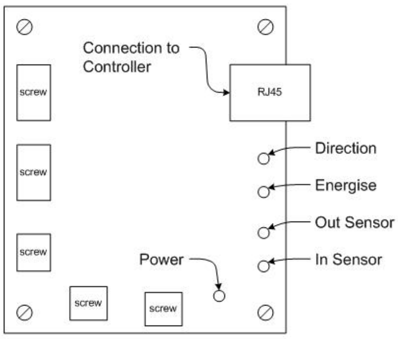
See appendix H for a description of pwm (pulse width modulation).

6.3 Conveyor

The orientation of the in and out sensors discussed below is defined relative to the conveyor's forward direction, *which is toward its motor.*

6.3.1 Indicators and Connectors

Indicator	Meaning
Direction	Off ⇒ Forward
	On ⇒ Reverse
Energise	On ⇒ Running
	Off ⇒ Stopped
Out Sensor	On ⇒ Sensor clear
	Off ⇒ Object present
In Sensor	On ⇒ Sensor clear
	Off ⇒ Object present
Power	On ⇒ Powered up



6.3.2 Channel Input and Output

Channel	Role	Mode	Logic
1	In Sensor	in	0 ⇒ present
2	Out Sensor	in	0 ⇒ present
3	Motor	out	1 ⇒ on
4	Direction	out	0 ⇒ forward

6.3.3 Changing Direction

The conveyor's direction should not be changed while it is moving. If it is in motion, then the recommended procedure is as follows:

1. Stop the conveyor: 0 → Motor
2. Change direction: 1 or 0 → Direction
3. Wait 0.5sec
4. Start the conveyor: 1 → Motor

A ARM Controller Bit Mapping

The relationship between the ARM I/O ports, the Olimex LPC-2378-STK board EXT2 header pins [3] and the controller interface is as follows:

Role	ARM Port	EXT2 Pin	Mode	RJ45 pin	Logic
Data 1	P2.7	30	out	–	1 ⇒ on
Data 2	P2.6	31	out	–	1 ⇒ on
Data 3	P2.5	32	out	–	1 ⇒ on
Data 4	P2.4	33	out	–	1 ⇒ on
Channel 1	P2.3	34	in/out	1	Depends on device
Channel 2	P2.2	35	in/out	3	Depends on device
Channel 3	P2.1	36	in/out	4	Depends on device
Channel 4	P2.0	37	in/out	7	Depends on device
5V	–	38	–	–	–
3.3V	–	39	–	–	–
GND	–	40	–	–	–

B RJ45 Device Adapter Link

Two pin configurations have been defined for RJ45 device adapter to controller interface links:

High Speed Four channels each using a standard RJ45 twisted pair [2]. This is the preferred option for four or less signals.

Low Speed Seven channels with common ground on pin 1.

The pins for these configurations are allocated as follows:

RJ45	Color	Pair	High Speed	Low Speed
1	White/Green	a	Channel 1	GND
2	Green	a	GND	Channel 1
3	White/Orange	b	Channel 2	Channel 2
4	Blue	c	Channel 3	Channel 3
5	White/Blue	c	GND	Channel 4
6	Orange	b	GND	Channel 5
7	White/Brown	d	Channel 4	Channel 6
8	Brown	d	GND	Channel 7

Signal levels on the link are 0-5V, with CMOS termination characteristics [5].

All of the cell's device adaptors use the high speed configuration. The channel allocations for each of the adaptors is give in §6.

C CAN RJ45 Connector

The configuration given by the CANopen standard [1] for RJ45 is as follows:

Pin	Signal	Description
1	`CAN_H`	Dominant High
2	`CAN_L`	Dominant Low
3	`CAN_GND`	Ground
4	Reserved	Upgrade Path
5	Reserved	Upgrade Path
6	`CAN_SHLD`	Shield, Optional
7	`GND`	Ground, Optional
8	`CAN_V+`	Power, Optional

D CAN 9-Pin D Connector

The configuration as given by the CANopen standard [1] for 9-pin D connectors is as follows:

Pin	Signal	Description
1	`-`	Reserved
2	`CAN_L`	Dominant Low
3	`CAN_GND`	Ground
4	`-`	Reserved
5	`CAN_SHLD`	Shield, Optional
6	`GND`	Ground, Optional
7	`CAN_H`	Dominant high)
8	`-`	Reserved
9	`CAN_V+`	Power, Optional

E Driver IC Replacement

The driver integrated circuits on the device adapter and interface boards can be damaged by incorrect configuration, or by shorting them to power or ground. In this case, they have to be replaced with new 74HC125 [4], 14 pin DIP (SOT27-1) packages [6]. The identities of the vulnerable integrated circuits are as follows:

PCB	Label
Alarm	IC2
Conveyor	IC2
Interface	IC1

F Resetting ARM Controllers

There can be a problem with the ARM boards caused by code turning on the PLL (Phase Locked Loop) using "strange" settings. This makes it impossible for the JTAG to halt the processor prior to writing into Flash or SRAM - the manual warns against this. The PLL is part of the NXP Si, and is used to provide very much higher clock rates than the XTL clock (12 MHz).

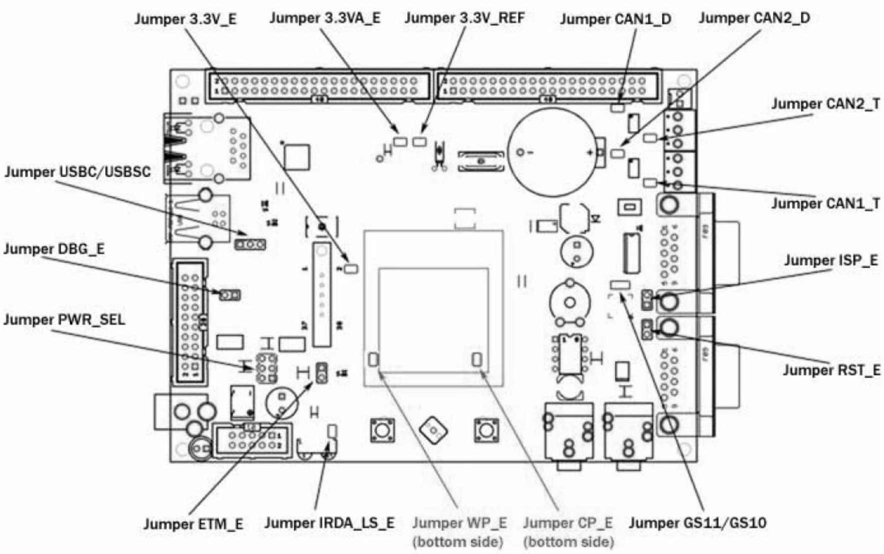
The NXP LPC2378 processor (and others in the same family) can be programmed via a serial post using what NXP call ICSP (In-Circuit Serial Programming). NXP provide a tool, *FlashMagic*, to program the ARM via a serial port using ICSP.

FlashMagic should be part of the standard PC image in the laboratory. The program is also available from `www.flashmagictool.com`. It is free for non-commercial use and is easy to install. To use it to reset the ARM board, proceed as follows:

- Prepare the board:
  - Remove the two retaining screws on the ARM board RS232 port 0 (the port nearest the front of the board) - this allows clear access to the D socket.
  - Place jumpers on `ISP_E` and `RST_E` to enable ICSP on the RS232 port 0.
- Power and comms:
  - Power up the board using the usual JTAG device from an available USB on your PC (no need to run EWARM at this stage)
  - Use a USB/RS232 protocol converter to connect the ARM RS232 (port 0) to your PC. You may have an available RS232 on your PC, in which case there will be no need for the protocol converter.
- Run FlashMagic:
  - From the programming interface, use the following settings:
    - Select Device: LPC2378
    - Serial port: COM11 - may be different for your USB/RS232 protocol converter
    - Baud rate: 19200
    - Interface: None (ISP)
    - Oscillator: 12 (MHz)
    - Check the box for "Erase all flash"
    - All other boxes are empty
  - Click on "Start"

    The erase should be very quick - the tool reports success.
- Test the board with EWARM to download using JTAG.

G ARM Configuration



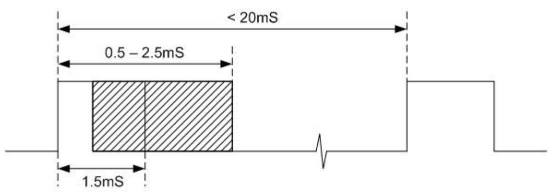
LPC-2378STK jumpers description, Olimex
http://www.olimex.com/dev/lpc-2378stk.html

Jumper name	Default	Position	Description
PWR_SEL			External power supply.
			+5V power from JTAG.
			+5V power supply from USB.
DBG_E			Disable TRACE port (P2.0 2.9).
			Enable TRACE port (P2.0 2.9).
USBC/USBSC			R27 (1.5k) is connect to USB D+
			R27 (1.5k) is controlled from P0.14/U2CONNECT/SSEL1 port (PIN48)
			P0.14/U2CONNECT/SSEL1 port (PIN48) is not connected.
ISP_E RST_E			Disable ICSP programming
			Enable ICSP programming (via RS232_0)

H Servo Pulse Width Modulation

Model servos are controlled using a form of Pulse Width Modulation (PWM). In general, the pulse width is between 0.5 and 2.5mS, and the period for pulse delivery is about 20ms, as shown in figure below. The smaller value positions the servo arm at its maximum counter-clockwise position, and the large value at its maximum clockwise position. A width of 1.5mS puts the arm in its central position. The period does not have to be precise and can vary but 20mS is the maximum.

However, permissible pulse widths depend on the make and model of servo. For example the Hitec servos pulse duration is from 0.9 to 2.1mS with 1.5mS as centre. Going outside the limits can damage the servo. If the servo's parameters are unknown, probably safe values are 1 to 2 ms.



References

[1] *CANopen DR-303 Cabling and Connector Pin Assignment*, Version 1.0, CAN in Automation (CiA), 1999.

[2] *Commercial Building Telecommunications Cabling Standard, Part 1: General Requirements*, TIA/EIA-568-B.1, May 2001.

[3] *LPC-2378STK development board Users Manual*, Rev C, Olimex Ltd, July 2009. `http://www.olimex.com/dev/pdf/ARM/LPC/LPC-2378STK.pdf`

[4] *74HC/HCT125 Quad Buffer/Line Driver with Active LOW Output Enable*, Philips Semiconductors, 1990. `http://ics.nxp.com/products/hc/datasheet/74hc125.74hct125.pdf`

[5] *74HC/T High-Speed CMOS User Guide*, Philips Semiconductors, Nov 1997. `http://ics.nxp.com/support/documents/logic/pdf/user.guide.hcmos.pdf`

[6] *74HC/T Package Outline Drawings Package Information*, Philips Semiconductors, 2003. `http://ics.nxp.com/support/documents/logic/pdf/family.hcmos.package.outlines.pdf`