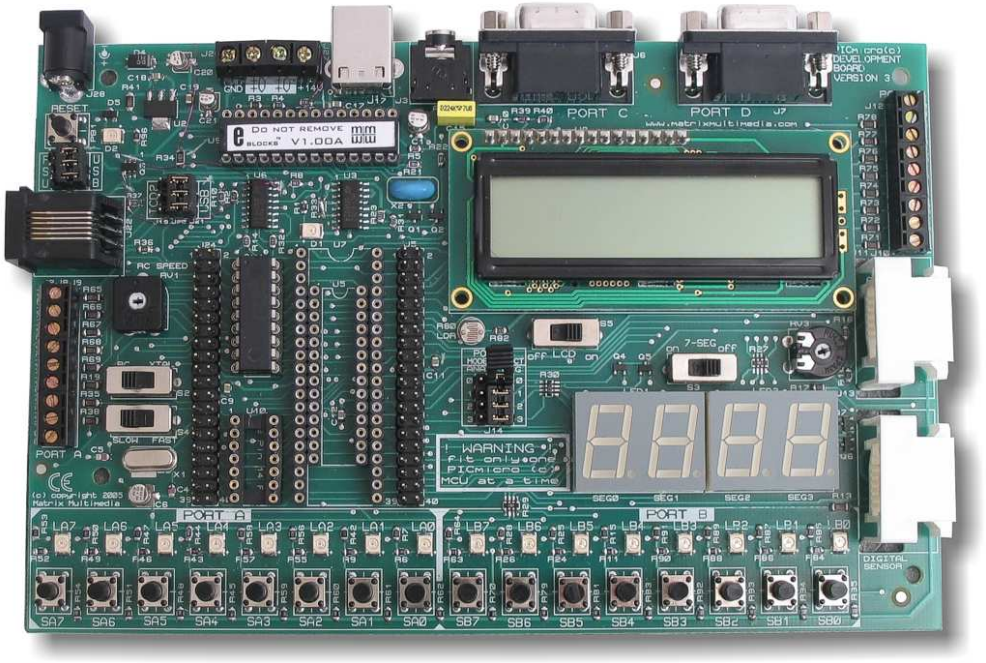
Version 3 PICmicro MCU development board
HP488-00-3
Technical datasheet



Contents

1. About this document

This document concerns the E-blocks PICmicro Microcontroller development board code HP488 version 3.

The order code for this product is HP488.

1. Trademarks and copyright

PIC and PICmicro are registered trademarks of Arizona Microchip Inc.
E-blocks is a trademark of Matrix Multimedia Limited.

2. Other sources of information

There are various other documents and sources that you may find useful:

Getting started with E-Blocks.pdf

This describes the E-blocks system and how it can be used to develop complete systems for learning electronics and for PICmicro programming.

PPP Help file

This describes the PPP software and its functionality. PPP software is used for transferring hex code to a PICmicro microcontroller.

C and assembly strategies

Not provided for this product.

3. Disclaimer

The information in this document is correct at the time of going to press. Matrix Multimedia reserves the right to change specifications from time to time. This product is for development purposes only and should not be used for any life-critical application.

4. Technical support

If you have any problems operating this product then please refer to the troubleshooting section of this document first. You will find the latest software updates, FAQs and other information on our web site: www.matrixmultimedia.com . If you still have problems please email us at: support@matrixmultimedia.co.uk.

2. General information

1. Description

This flexible development board is an ideal platform for learning and project development.The board will program a range of 8, 14, 18, 28 and 40 pin PICmicro® microcontroller devices from the 12, 16 and 18 series PICmicro range. The board is programmed using the USB port (from which it can also take power) and is supplied with a comprehensive programming utility - PPP. The board is ideal for using in conjunction with one of our 3 PICmicro CD ROM resources to help learn PICmicro programming using flow charts, C or assembly code. The board can program Low Voltage Programmable PICmicro MCUs and deliver a limited amount of power using only the USB supply, but we recommend that an external power supply (product code HPPSU2) is used to take maximum advantage of the board's features. The board is compatible with our range of E-blocks modules which contain circuitry for a wide range of functions - from simple switches and LEDs, through to internet servers, and Bluetooth transceivers. The board is also compatible with Microchip's In Circuit Debugging 2 system.

2. Features

- Makes it easier to teach and learn PICmicro programming
  - Designed for educational use
  - 3 CD ROM based resources in assembly, C, and flow chart programming are available
  - Free download software, PPP, provides seamless send and verify functions

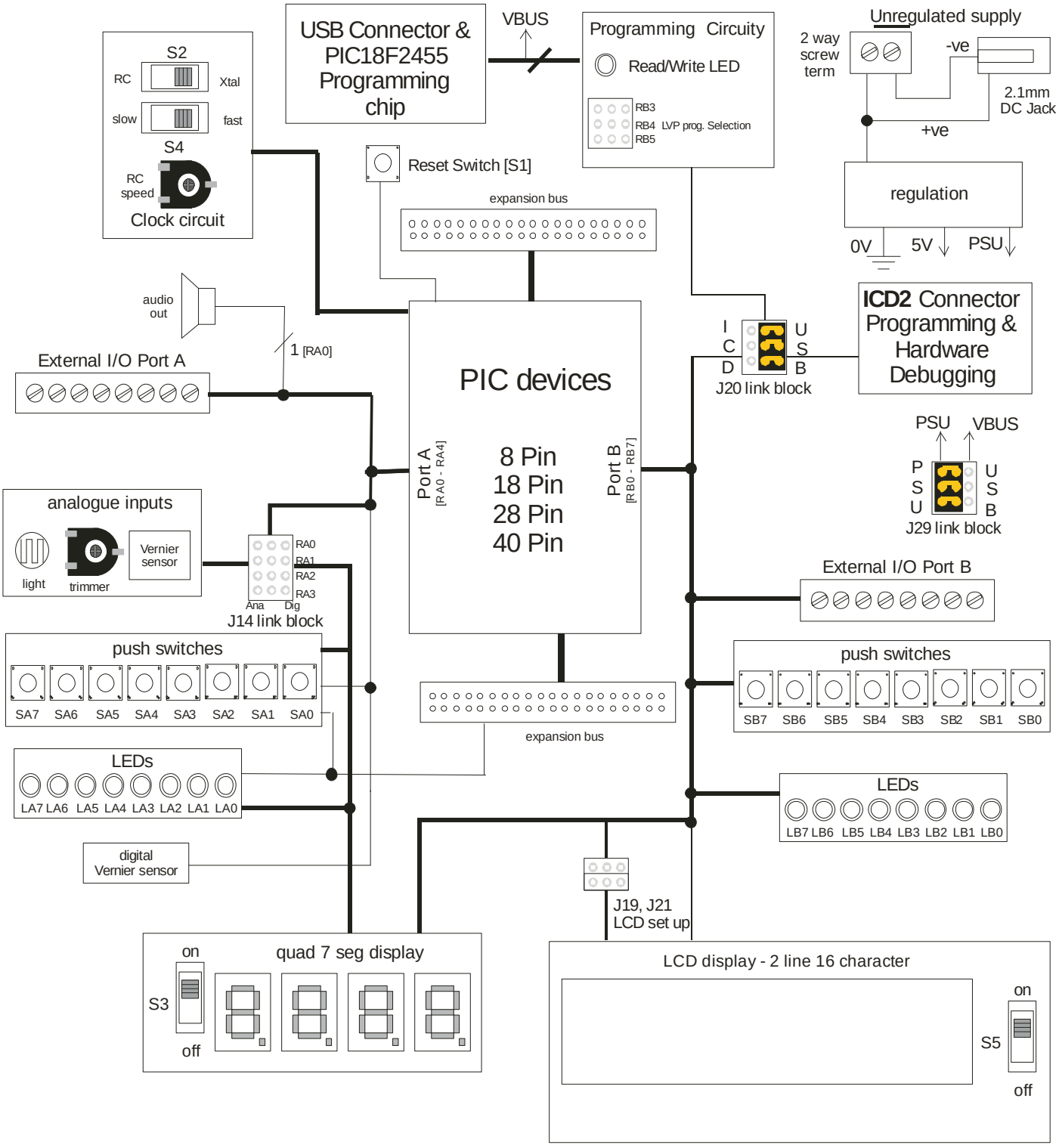
- Makes it easier to develop PICmicro projects
  - Supports low cost Flash-programmable PICmicro devices
  - Fully featured displays - 16 individual LEDs, quad 7-segment display and LCD display
  - USB programmer
  - Supports PICmicro microcontrollers with A/D converters
  - On-board sensors
  - A comprehensive range of external analogue and digital sensors are available for project work
  - Fully protected expansion bus for project work
  - All inputs and outputs available on screw terminal connectors for easy connection
  - E-Blocks compatible with 2 E-Blocks connectors fitted to ports C and D
- Low cost
- Used as a programmer and as a development board
- Programs a wide range of PICmicro MCU devices
- Full suite of programming software available
- RC or Xtal operation
- In-Circuit Debugging via MPLAB® ICD2

3. New features for Version 3

The following are the improvements made on version 3:

- Improved power circuit for either polarity power supply
- Compatible with a wider range of devices in Low Voltage Program mode
- Greater options for LCD connection to port B for Low Voltage Programming
- Support for a wider range of PICmicro devices
- Fast programming – 8k in under 10 seconds

4. Block schematic



5. Supported devices

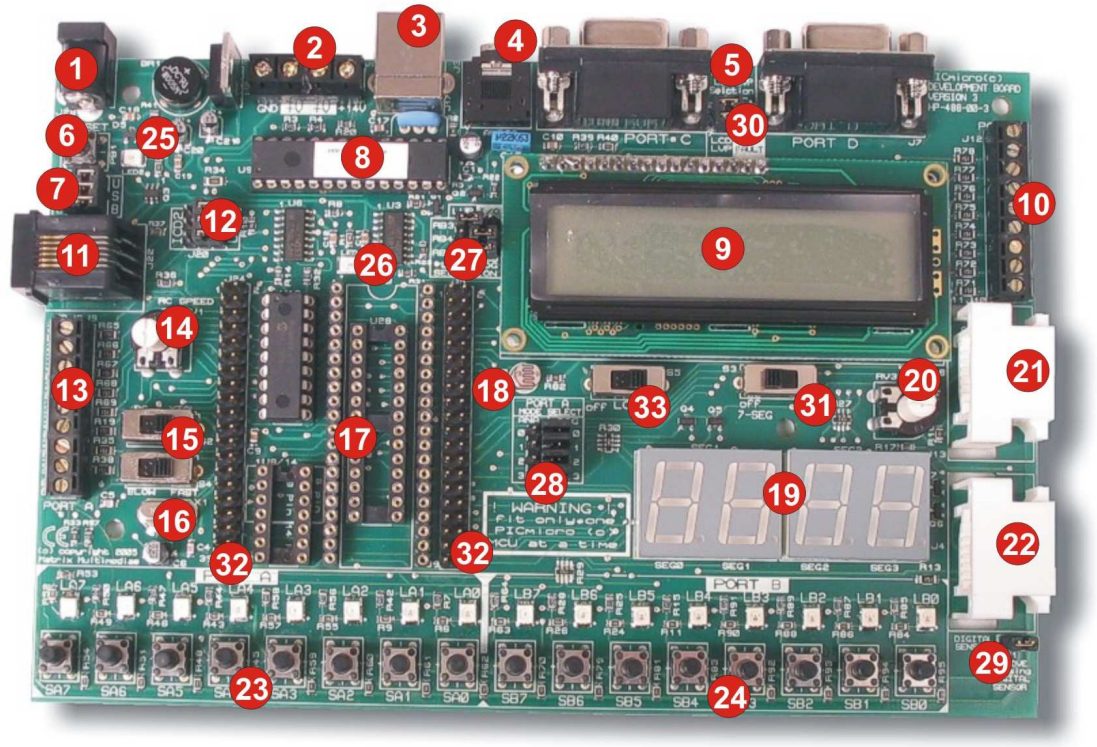
Currently PPP and the HP488 support the following devices:

PIC12F629, PIC12F675, PIC12F635, PIC12F683

PIC16F627A, PIC16F627, PIC16F628A, PIC16F628, PIC16F630, PIC16F648A, PIC16F676, PIC16F684, PIC16F688, PIC16F636, PIC16F716, PIC16F72, PIC16F737, PIC16F73, PIC16F747, PIC16F74, PIC16F767, PIC16F76, PIC16F777, PIC16F77, PIC16F818, PIC16F819, PIC16F83, PIC16F84A, PIC16F84, PIC16F870, PIC16F871, PIC16F872, PIC16F873A, PIC16F873, PIC16F874A, PIC16F874, PIC16F876A, PIC16F876, PIC16F877A, PIC16F877, PIC16F87, PIC16F88

PIC18F242, PIC18F248, PIC18F252, PIC18F258, PIC18F442, PIC18F448, PIC18F452, PIC18F458, PIC18F1220, PIC18F1320, PIC18F2220, PIC18F2320, PIC18F2331, PIC18F2410, PIC18F2420, PIC18F2431, PIC18F2439, PIC18F2455, PIC18F2510, PIC18F2515, PIC18F2520, PIC18F2525, PIC18F2539, PIC18F2550, PIC18F2585, PIC18F2586, PIC18F2610, PIC18F2620, PIC18F2680, PIC18F2681, PIC18F4220, PIC18F4320, PIC18F4331, PIC18F4410, PIC18F4420, PIC18F4431, PIC18F4439, PIC18F4455, PIC18F4510, PIC18F4515, PIC18F4520, PIC18F4525, PIC18F4539, PIC18F4550, PIC18F4585, PIC18F4586, PIC18F4610, PIC18F4620, PIC18F4680, PIC18F4681

3. Board layout



HP488-74-3.cdr

1. Power supply connector
2. 5V outputs for E-blocks connection
3. USB connector
4. Single bit audio output
5. E-blocks connectors for ports C and D
6. Reset switch
7. Power selector link block—USB or Power connector
8. USB interface chip
9. 2 line 16 character alphanumeric display
10. Screw terminal connectors for port B
11. In Circuit Debug socket
12. ICD/USB programming selection
13. Screw terminal connectors for port A
14. RC speed potentiometer
15. Oscillator mode selection switches
16. Removable crystal
17. Sockets for programming 8, 14, 18, 28 and 40 pin PICmicro devices
18. On-board light sensor
19. Quad 7-segment displays
20. Analogue input potentiometer
21. External analogue sensor input
22. External digital sensor input
23. Port A switches and LEDs
24. Port B switches and LEDs
25. Power indicator
26. Programming indicator
27. LVP pin selector
28. Port A analogue/ digital selector
29. Digital sensor selector
30. LCD display LVP mode selector
31. 7-segment display enable
32. Expansion bus
33. LCD enable

4. Testing this product

The following program will test the circuit. The test file can be downloaded from www.matrixmultimedia.com.

1. installing PPP

To install run PPPv3.exe, which is located at <D>:\PPPv3\PPPv3.exe and follow the instructions provided. *<D> refers to your CD drive.*
By default PPPv3 is installed into:
C:\Program Files\Matrix Multimedia\Common\PPPv3\

There are four 'Features' that can be installed.

- PPP v3 core files - this is PPP v3 itself, and should be installed.
- Update ASM4PICs - This feature allows you to update ASM4PICs to use PPP v3.
- Update C4PICs - This feature allows you to update C4PICs to use PPP v3.
- Update Flowcode - This feature allows you to update Flowcode to use PPP v3.

The three updates will be automatically installed. Select the 'X' 'Do not install option' if you do not to update a feature.
If you need to update a product at a later date you can re-run the install and update that feature.

There is more help and information available on the CD provided at <D>:\PPPv3\readme.txt
<D> refers to your CD drive.

When you connect the Multiprogrammer to your computer, via the USB cable, the first time there will be installation routine for this 'new hardware'. This for most users will be a 'plug and play' routine where your computer will automatically recognize the hardware.

Running on Windows 2000 / ME / XP

These programs allow 'plug and play' for your new hardware. Therefore when you first connect the Multiprogrammer to your computer you will receive a pop-up screen that indicates that there is new hardware connected to the computer. The program itself will deal with any installation of any drivers that it requires internally. Therefore you can use your Multiprogrammer immediately.

Running on Windows 98 (you will need the Windows 98 CD at hand)

When you connect the Multiprogrammer to the computer your Windows 98 program will run a 'New Hardware Wizard'. This procedure is straightforward and easy to understand. Follow the on-screen instructions. Once this has been completed the Mulitprogrammer will be ready to use.

There is more detailed information and help on the CD provided at <D>:\eblocks\Installation Guide.doc

2. System Setup – testing with an external power supply

Set up development board with:

HP488 Options	Setting
Power supply	External, 14V
PICmicro device	16F88
S4 (Fast/Slow)	Don't care
S2 (RC/Xtal)	Xtal
Xtal frequency	19.6608MHz
Port C	
Port D	
Test program	f_test_1.hex

3. Testing the board – with an external power supply

1) Ensure power is supplied to the board
    1. USB cable required
    2. PSU cable required
2) Set Jumper J29 to 'PSU
3) Ensure Jumper J20 set to USB
4) Send the test program to the board using PPP
5) Check the illumination of all Port A and B LEDs

4. Testing the board – with a USB power supply

The following instructions explain the steps to test and use your board in low voltage programming mode. Microchip® have enabled this feature is some devices however it has some adverse effects on the function of the PIC chips.

Follow these instructions to program the board in Low Voltage Programming (LVP) mode.

Hardware Set-up

Set up development board with:

HP488 Options	Setting
Power supply	External, 14V
PICmicro device	16F88
S4 (Fast/Slow)	Fast
S2 (RC/Xtal)	RC
Xtal frequency	Don't care
Port C	
Port D	
Test program	f_test_2.hex

1) Ensure power is supplied to the board
    1. USB cable required
2) Set Jumper J29 to 'USB'
3) Ensure Jumper J20 set to USB
4) Send the test program to the board using PPP
5) Check the illumination of all Port A and B LEDs

Note

If the chip has been set to high voltage programming then the board will require a high voltage to re-enable the LVP. This must be done before attempting to program in LVP – however when shipped the chip is already configured in LVP.

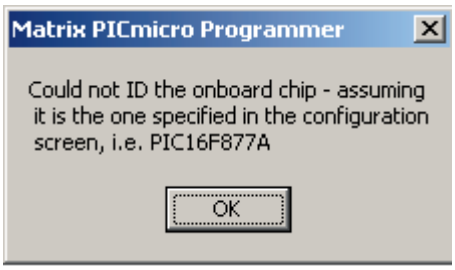
5. Important information regarding LVP

When using LVP, bit 3 of Port B on the PIC16F88 is not functional as an I/O line as it is part of the LVP programming architecture set by Microchip®.

6. Trouble shooting for Low Voltage Programming

Due to the internal architecture of the Microchip® PICmicros a high voltage is needed to re-enable Low Voltage Programming (LVP) mode. The following instructions indicate how to tell if the chip is not in LVP mode and how to re-enable this feature.

Indicating if the PICmicro is in LVP mode or not
If LVP is disabled the programming software PPP will not be able to ID the on board PICmicro. A similar pop-up screen will appear: -



Then the PPP will indicate that the chip has not been erased. There is no point in continue to try to program the PICmicro.

Re-Enabling the LVP function
This requires the use of an +13.5V external power supply (see Section 6 for details).
In order to re-enable the PICmicro must be erased. This erase function returns the PICmicro to the factory settings and thus LVP is enabled.

To Erase the PICmicro to enable LVP mode following these instructions: -

1) Remove all power supplies to the board
2) Remove USB cable from socket
3) Remove external power supply from socket
4) Place jumper J29 to 'PSU' (left-hand side)
5) Insert USB cable into socket (J1)
6) Insert +13.5V power supply via socket
7) In PPP click File -> Erase PICmicro
8) Note this cannot be done directly from Flowcode

The PICmicro will now be enabled for LVP.

5. Circuit description

The PICmicro development board is designed for learning how to program PICmicro applications in educational and industrial context. It also allows a wide range of PICmicro microcontrollers to be programmed and the 'seamless' nature of the programming software supplied with the product ('PPP') makes it suitable for the development of a range of PICmicro projects.

The board is optimised for use with a PIC16F88 from Arizona Microchip, which has a number of features:

- Crystal or RC operation
- 2 ports: Port A - 5 pins and Port B - 8 pins i.e.: 15 programmable pins that can all be used as inputs and outputs.
- Flash programmable - up to 1,000,000 programming cycles
- On-board A/D, internal clock, internal USART and a host of other functions and features

The PIC16F88 has analogue capability and is a very functional PICmicro. It has been selected for use with this development board for a number of reasons:

- It is a low cost device - suitable for project work in schools
- It is electrically re-programmable
- Features are up-ward compatible with larger chips – such as the PIC16F877A

A full list of other PICmicros that the board can support is listed below.

The board has been designed to allow those with little or no experience of embedded microcontroller programming to be able to produce highly functional designs in as short a time as possible. With this in mind we have included switches on port A and B input/output pins, a quad 7-segment display, a 2 line 16 character alphanumeric display, a choice of crystal or RC oscillator, simple on-board analogue sensor (light) and sensor simulator (potentiometer) and the ability to interface the board to a host of external sensors including motion, temperature, pH. humidity etc. Ports A and B are also connected to screw terminals on the sides of the board via 150ohm resistors which protect the device on the board itself.

The board can be used with ZIF sockets if required: please bear in mind that you may need to insert one or more turned pin DIL sockets to raise the height of some ZIF sockets to clear surrounding components.

Full tutorials in programming the PICmicro in Assembly code, C, and flowcharts are available. These sets of tutorials are available on CD ROM and require no other software - all compilers and development environments are included on the CDs themselves.

1. Schematic description - please refer to block schematic

The board is based on a flexible topology that allows the inputs and outputs to be used in a number of ways depending on switch settings on the board. This flexibility means that students will, as they use additional features, need to be aware of how a single input-output (I/O) pin can be used for a number of functions dictated by the circuitry attached to it. This is explained in the Circuits section of this document.

The architecture of the board is split into two main buses - the port A bus and the port B bus. Each line on each bus can be used as either an input or as an output. The connectors and displays on the board are all wired in parallel on the PIC pins for convenience but clearly connecting more than one device to one pin will result in conflict: **students will need to be made aware of this.**

2. Port A bus

Under normal operation all of the jumper links in the J14 link block are in the Digital position. This means that the Port A I/O pins are routed to the push switches and LEDs. The actual circuit is described in the Circuits section. Note that operating the push switches whilst a voltage is connected to port A via the external screw terminals is not advisable, as it will result in unpredictable results. Port A screw terminal connectors are fed to the pins on the PICmicro via 150 ohm resistors which provide short circuit protection.

With J14 links in the Digital position port A is routed to its push switches (SA0 to SA7), LEDs, digital Vernier sensor I/O, and the quad 7-segment display. Bits RA0, RA1, RA2 and RA3 are used for the common anodes of each of the four 7-segment displays. RA0 is used for tone generation via the jack plug, and bit RA2 is used for the digital Vernier sensor input.

With J14 links in the Analogue position port A is switched to the analogue sensor section of the board. This means that RA0 is connected to the on-board light sensor, RA1 is connected to the potentiometer resistor to form an adjustable voltage divider for sensor simulation, and bit RA3 is connected to the external analogue Vernier sensor connector.

With J14 links in the Analogue position you should be aware that on-board switches and LEDs RA0-3 will not operate.

If you want to use the external port A screw terminal connectors for analogue inputs you should remove the links off J14 altogether. Note that links can be set individually so that RA0 could be used in analogue mode and the rest in digital mode.

3. Port B Bus

Port B I/O pins are routed to its push switches (SB0 to SB7), the LEDs (LB0 to LB7), the quad 7-segment displays, the LCD display and the screw terminal connectors. Note that the screw terminal connectors are fed to the pins on the PICmicro via 150ohm resistors, which provide short circuit protection. However operating the push switches whilst a voltage is connected to Port B is not advisable as it will result in unpredictable results.

Port B is also used for programming the PICmicro - the PPP software used for programming the PICmicro will automatically take over pins required for programming, verifying and resetting the PICmicro on the board.

If you are using the board in LVP mode then B3 (or B4 or B5 for some chips) may not be used as an I/O line.

4. Port C & D E-Block Bus Connection

Ports C and D of the PICmicro MCU are fed out onto 9 way D-type connectors at the back of the board. These can be used to add one of a wide range of E-blocks boards to the development board. E-blocks vary from simple LED and switch boards through to more complex subassemblies like Bluetooth boards, CAN bus, internet boards, IrDA transceivers etc.

The addition of E-blocks boards allows you to considerably expand the functionality of your system to develop a very wide range of electronic systems.

5. Port E Bus Connection

Port E is available on larger PICmicro MCU devices. This is not a full port (only three bits) and therefore most users will not need to use this port. This port is available on the IDC header pins (J5 and J24). The details of the actual pins on this header can be found in the “Pin comparison chart” section of this document.

6. Clock circuitry

During early stages of learning PICmicro programming students often want each clock cycle to be large (1 second) so that they can see code executing very slowly (flashing LEDs don’t simply become ’less bright’ LEDs). The PIC16F88 supports this feature by having an RC mode of operation.

Use of RC mode also reduces component count and cost for small projects.

S2 selects between RC mode and Crystal mode.

S4 selects fast or slow operation when RC mode is selected.

RV1 adjusts the speed of the clock within a range dictated by S4.

However more advanced students require a faster clock speed, for setting up parameter such as the LCD module. Therefore the board is supplied with a 19.6608MHz crystal. The crystal is supplied mounted in ultra low profile sockets – allowing the crystal to be changed.

7. Power supply circuit

The board is normally operated from a regulated DC supply of 13.5V. This allows full operation including programming.

The board can be operated in Low-voltage mode via solely the USB cable provided. However care must be taken, as there is only limited power that can be taken from a computers USB port. Also only certain chips can be reprogrammed in this Low-voltage mode, refer to the specific chip datasheet to determine if it has LVP function. The PIC16F88 provided with this board has been set-up to accept low-voltage programming (LVP), but some features are not available in this mode, such as LB3. If this mode is disabled then the 13.5V regulated DC power supply must be used, and only using this high-voltage programming can the low-voltage programming mode be re-enabled. Please refer to the specific PIC datasheet regarding LVP function.

Please note that not all chips have the Low-Voltage Programming function and therefore these chips must be programmed using an external power supply as stated above.

The jumper link system, J29, allows the user to decide on the source of the power supply. If using a regulated 13.5V power supply the jumper should be positioned to the left hand side of the jumper system labelled 'PSU'. If using USB power place the jumper on the right hand side of the jumper system. The jumper should always be orientated so that the 3 links in the jumper block are always position horizontally – thus connecting the centre pins to either the left hand or right hand pins. LED2 indicates that power is supplied to the board from either the external power supply or the USB cable.

Please note that both USB and the PSU cables should be removed for the Multiprogammer board BEFORE changing the position of this jumper.

When using the 13.5V regulated power supply the board will only supply up to 250mA. This is due to the thermal dynamics of the on board regulator. Therefore if more current is required the DC supply voltage must be reduced to 8 – 9V and LVP mode must be used.

Remember that other E-blocks will have to receive 5V by placing a connecting wire from the "+V Out" screw terminal of the Multiprogrammer to the "+V" screw terminal of each E-Block that requires a voltage.

8. Displays

The LCD display is connected to Port B I/O. The LCD display is turned on by switch S5.

The quad 7-segment display is connected to both Port A and Port B. Port B is used to control each of the 8 segments (7 for the main character and 1 for the full stop). Port A bits 0 to 3 are used to select each of the 4 characters. The quad 7-segment display is turned on by switch S3.

Note that turning on the display will affect the operation of Port B I/O and when using the quad 7-segment displays it is recommended that you do not use Port B for any other purposes. Under certain circumstances you will notice that segments of the quad 7-segment display may be lit up very dimly by Port A/B operation even when S3 is in the off position. This is a feature of the low cost driver circuitry and is unavoidable.

The jumpers J19 and 20 can be used to select alternative pins for the LCD where Low Voltage Programming is required. J19 selects D7 for the LCD from RB3 (default) or RB6. J20 selects the RS pin for the LCD from RB5 (default) or RB7. Due to space constraints the LCD display can not be used with devices that require RB4 to be used as the LVP pin.

9. Microchip ® In-Circuit Debugger (ICD2)

The board is compatible with the Microchip® ICD2 in-circuit debugging tool. This allows the debugging of the actual hardware and the software program that is programmed onto the chip. The ICD2 allows the user to step through the program whilst it is actually programmed on the chip – therefore allowing the hardware to be tested with the software. The ICD2 can also program the targeted chip as well. For more information regarding the ICD2 please visit the Microchip website: www.microchip.com

The ICD2 for Microchip® plugs directly into the ICD socket (J22) on the Development board. Jumper J20 must be moved to the left hand side of the pins, into the 'IDC2' position. This means that the ICD2 can now control the programming and the in-circuit debugging of the hardware. Note that the ICD2 has speed limitations and ifusing an ICD2 you should use a crystal of less than 5MHz.

If using an ICD2 then the power must be applied to the ICD2 before power is applied to the development board.

10. J18 Low Voltage programming pin selector – Caution

J15, 16, 18 are to the left hand side of the LCD display with the legends hidden by the LCD display board.

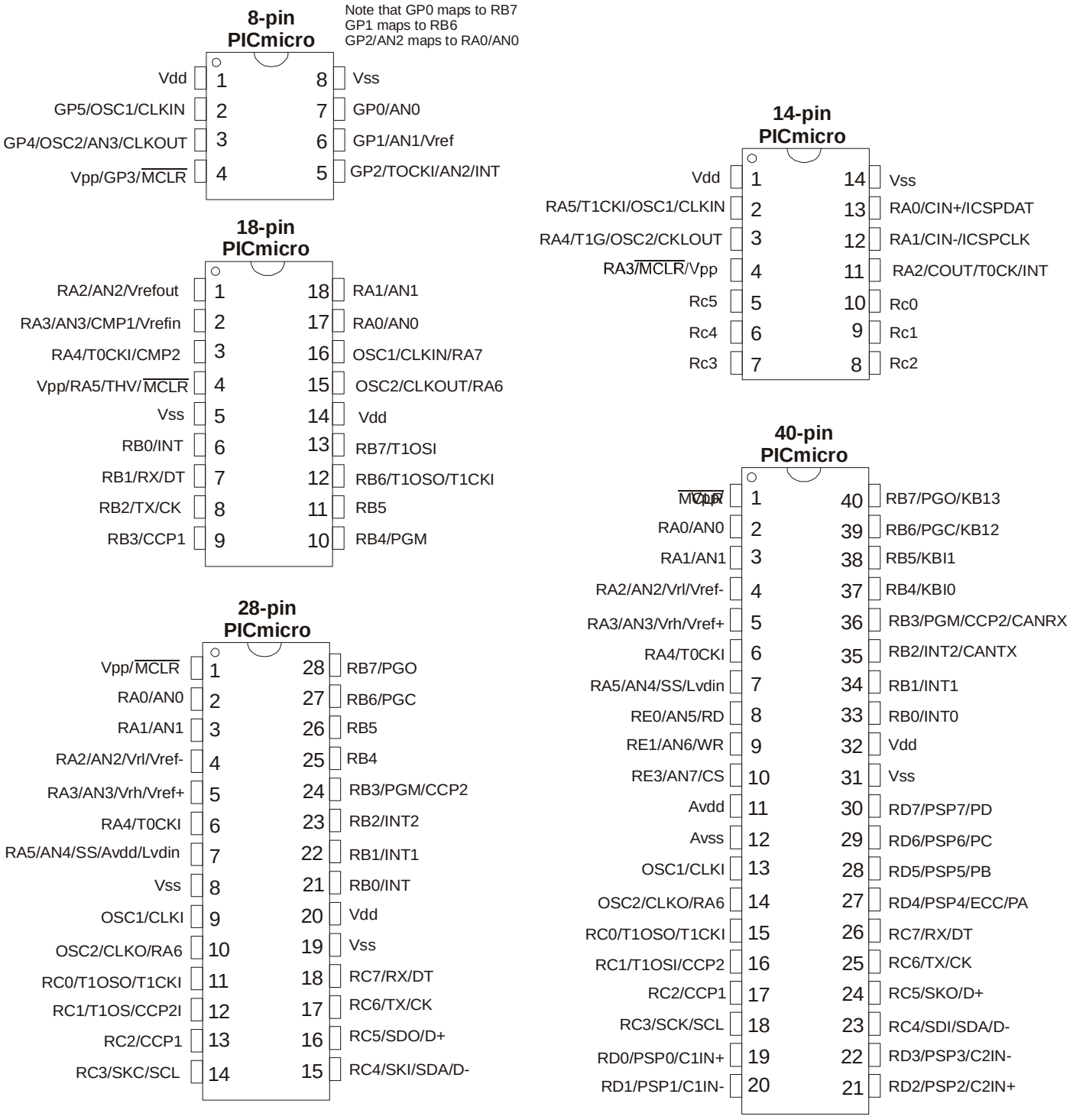
A large number of PICmicro devices reserve pin RB3 as the Low Voltage Progamming selection pin. Some use pins RB4 or RB5. J15, 16, 18 enables you to select Low Voltage Programming and make a choice of whether you want the associated pin to be an I/O line or the LVP pin.

Note that only one of the three pins RB3 to 5 must be selected as the LVP pin – the left hand position If you place more than one link on the left hand pins then you will damage the target device.

If you wish to disable these links to prevent accidents then solder a wire across the appropriate link selection pins.

6. PICmicro microcontroller pin out details

Broadly speaking the ranges of PICmicro devices are designed to be upwards compatible: the pin functions on an 18-pin device are available on a 28-pin device and a 40-pin device. This can be seen from the following excerpt from the Microchip product selector card. The following diagram shows the pin out of the various PICmicro devices:



7. Bus connections and jumpers

1. Expansion bus

The pin connections on the expansion bus exactly mirror the pin numbering on the 40-pin DIL socket. Note that the pin numbering on the IDC socket is slightly different to that on a DIL socket which results in the seemingly odd arrangement of pins on the IDC pin chart.

Pin Comparison Chart

	PICmicro Pinout				
Bus Name	18 Pin	8 Pin	14 Pin	28 Pin	40 Pin
Vpp/MCLR	4	4	4	1	1
Vdd	14	1	1	20	11 & 32
Vss	5	8	11	8 & 19	12 & 31
OCS1	16	2	2	9	13
OCS2	15	3	3	10	14
RA0/AN0	17			2	2
RA1/AN1	18			3	3
RA2	1			4	4
RA3/AN3	2			5	5
	3			6	6
RA5/AN4	4			7	7
RB0	6			21	33
RB1	7			22	34
RB2	8	5*	11*	23	35
RB3	9			24	36
RB4	10			25	37
RB5	11			26	38
RB6	12	6*	12*	27	39
RB7	13	7*	13*	28	40
RC0			10	11	15
RC1			9	12	16
RC2			8	13	17
RC3			7	14	18
RC4			6	15	23
RC5			5	16	24
RC6				17	25
RC7				18	26
RD0					19
RD1					20
RD2					21
RD3					22
RD4					27
RD5					28
RD6					29
RD7					30
RE0/AN5					8
RE1/AN6					9
RE2/AN7					10

For the 18, 28, and 40 pin devices the buses on devices are largely upwards compatible – pin connections on an 18-pin device appear on a 28-pin device and a 40-pin device, and pins on a 28-pin device appear on a 40-pin device. This allows the 18, 28, and 40 pin DIL sockets to be connected in parallel with the PICmicro bus structure intact.

* This parallel connection is not possible with 8 and 14 pin devices due to programming requirements which means that there are anomalies with the pin connections for the 8 and 14 pin devices as follows:

Multiprogrammer port line	Connection pin on 14 pin device	14 pin port line
RB2	5	RA2
RB6	6	RA1
RB7	7	RA0

Multiprogrammer port line	Connection pin on 8 pin device	8 pin port line
RB2	1	GP2
RB6	12	GP1
RB7	13	GP0

2. Connections on J5

Bus Name	40 Pin	IDC connector
Vpp/MCLR	1	1
VCCchip	11 & 32	18 & 21
GND	12 & 31	20 & 23
OCS1	13	25
OCS2	14	27
RA0/AN0	2	3
RA1/AN1	3	5
RA2	4	7
RA3/AN3	5	9
RA4/AN4	6	11
RA5	7	13
RB0	33	16
RB1	34	14
RB2	35	12
RB3	36	10
RB4	37	8
RB5	38	6
RB6	39	4
RB7	40	2
RC0	15	29
RC1	16	31
RC2	17	33
RC3	18	35
RC4	23	36
RC5	24	34
RC6	25	32
RC7	26	30
RD0	19	37
RD1	20	39
RD2	21	40
RD3	22	38
RD4	27	28
RD5	28	26
RD6	29	24
RD7	30	22
RE0/AN5	8	15
RE1/AN6	9	17
RE2/AN7	10	19

Note J5 is a set to copy the 40-way DIL socket

3. Connections on J24

Bus Name	IDC connector
Vpp/MCLR	1
VCCchip	21 & 22
VCC	29

VPPmain	19
VBUS	17
GND	23 & 24
OCS1	26
OCS2	28
RA0/AN0	4
RA1/AN1	6
RA2	8
RA3/AN3	10
RA4/AN4	12
RA5	14
RA6	25
RA7	27
RC0	30
RC1	32
RC2	34
RC3	36
RD0	38
RD1	40
RE0	16
RE1	18
RE2	20
PRB0	15
PRB1	13
PRB2	11
PRB3	9
PRB4	7
PRB5	5
PRB6	3
PRB7	1
PRA0	31
PRA1	33
BRB3	35
BRB6	37
BRB7	39

8. Liquid Crystal Display

The LCD is a 16 character x 2 lines module. Internally it is 40 characters x 2 lines. Line 1 ranges from H'00' to H'27' and Line 2 ranges from H'40' to H'67'.

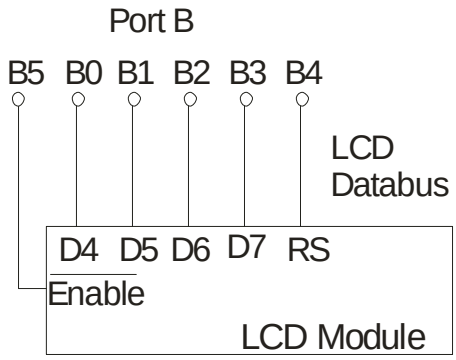
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67

Display Window [16x2]

Internal [40x2]

The LCD Module uses a Samsung KS0066U controller, which is similar to the Hitachi HD44780 controller.

The PICmicro board uses port B [B0 to B5] to program the LCD, as shown in the circuit diagram below. When the PICmicro board is turned on, data can only be sent to it after 30ms, this is the time taken for the LCD to initialize [as it clears all the RAM and sets up the Entry Mode]. Default connections are:



Note that you can use J19 and J21 to alter RS and Enable respectively.
To send a command to the LCD, data must be sent in two steps, the MSB followed by the LSB [byte is data on B0 to B3]. As each byte is sent to the LCD, B5 must be go high then low, for the LCD to acknowledge the byte. After the second byte has been acknowledged the LCD executes the command. The PICmicro board must wait for at least the length of the execution time for that command, before the next command can be sent. A timing diagram of this process is shown below.

Timing Diagram

The first command to be sent to the LCD must be 'Function Set' [to setup the LCD], this is usually followed by 'Display Control' and then 'Clear Display'. According to 'Entry Mode Set' after each character is sent to the LCD, the position of the cursor changes [by default it is incremented].

LCD Instruction Set

Instruction	Code										Description	Execution Time
MSB / LSB	B4	B3	B2	B1	B0							
Clear Display	0	0	0	0	0	0	0	0	1		Clear all display data. Set DDRAM address to 0. Move cursor to home position. Entry mode set to increment.	1.53 ms
Return Home	0	0	0	0	0	0	0	1	X		Set DDRAM address to 0. Move cursor to home position.	1.53 ms
Entry Mode Set	0	0	0	0	0	0	1	I/D	SH		Sets cursor move direction (I/D), specifies to shift the display (S). These operations are performed during data read/write.	39 us
Display Control	0	0	0	0	0	1	D	C	B		D is Display ON/OFF bit. C is Cursor ON/OFF bit. B is Blink Cursor ON/OFF bit.	39 us
Cursor/Display Shift	0	0	0	0	1	S/C	R/L	X	X		Sets cursor-move or display-shift (S/C), shift direction (R/L). DDRAM contents remains unchanged.	39 us
Function Set	0	0	0	1	0	1	0	X	X		Configuration data for setting up LCD. [Send First]	39 us
Set CGRAM Address	0	0	1	A5	A4	A3	A2	A1	A0		Sets the CGRAM address. CGRAM data is sent and received after this setting.	39 us
Set DDRAM Address	0	1	A6	A5	A4	A3	A2	A1	A0		Sets the DDRAM address. DDRAM data is sent and received after this setting.	39 us
Write Data to RAM	1	D7	D6	D5	D4	D3	D2	D1	D0		Writes data to CGRAM or DDRAM.	43 us

DDRAM is Display Data RAM
DDRAM address is location of cursor
CGRAM is Character Generator RAM
X is Don t Care

Bit Name	0	1
I/D	Decrement cursor position	Increment cursor position
SH	No display shift	Display shift
D	Display off	Display on
C	Cursor off	Cursor on
B	Cursor blink off	Cursor blink on
S/C	Move cursor	Shift display
R/L	Shift left	Shift right

Hint
please look at our web site for examples of code that shows how to program the 7-segment display. (TEST40.ASM factory test routine.)

LCD Character Set

9. Sensors

To use an analogue sensor, the Development board must manually be set to analogue mode and an analogue-capable PIC must be used. Whilst in analogue mode certain port A pins [A0/RA0,A1/RA1 etc] are able to convert an analogue signal to an 8 or 10 bit digital signal, using the on-board A to D converter inside the PICmicro device.

Please refer to Microchip data sheets for more details.

The analogue sensors are not normally connected to the PICmicro pins but are switched to them under the control of the jumpers on J14.

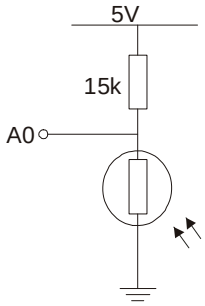
To use the external digital sensor please remove the link J1.

A full list of the 40 sensors compatible with this board can be found at www.matrixmultimedia.com.

1. LDR

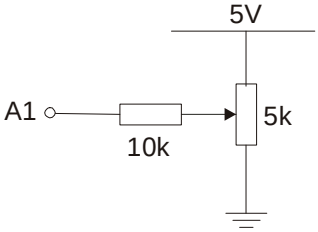
A compact cadmium sulphide light dependent resistor is soldered onto the development board. When J14 link RA0 is in the 'analogue' position it is connected to A0 on the PICmicro microcontroller. The resistance of this device reduces as light falling on to it increases.

Technical specification	
Dark resistance	1M Ohm
Resistance @ 10 Lux	10-20k Ohm
	100 Lux 2-4k Ohm
Peak spectral response	540 nm



2. Potentiometer

Most passive sensors are based on a simple potential divider circuit. In order to get your code up and running as simply as possible it is therefore useful to have a simulation of the full operation range (0V to 5V) that a sensor might provide. This function is simulated by a simple 4k7 preset potentiometer (RV3) with a series 10k resistor, which produces a voltage in the range of 0V to 5V on RA1. Note that J14 / 1 will need to be in the analogue position to use this circuit.



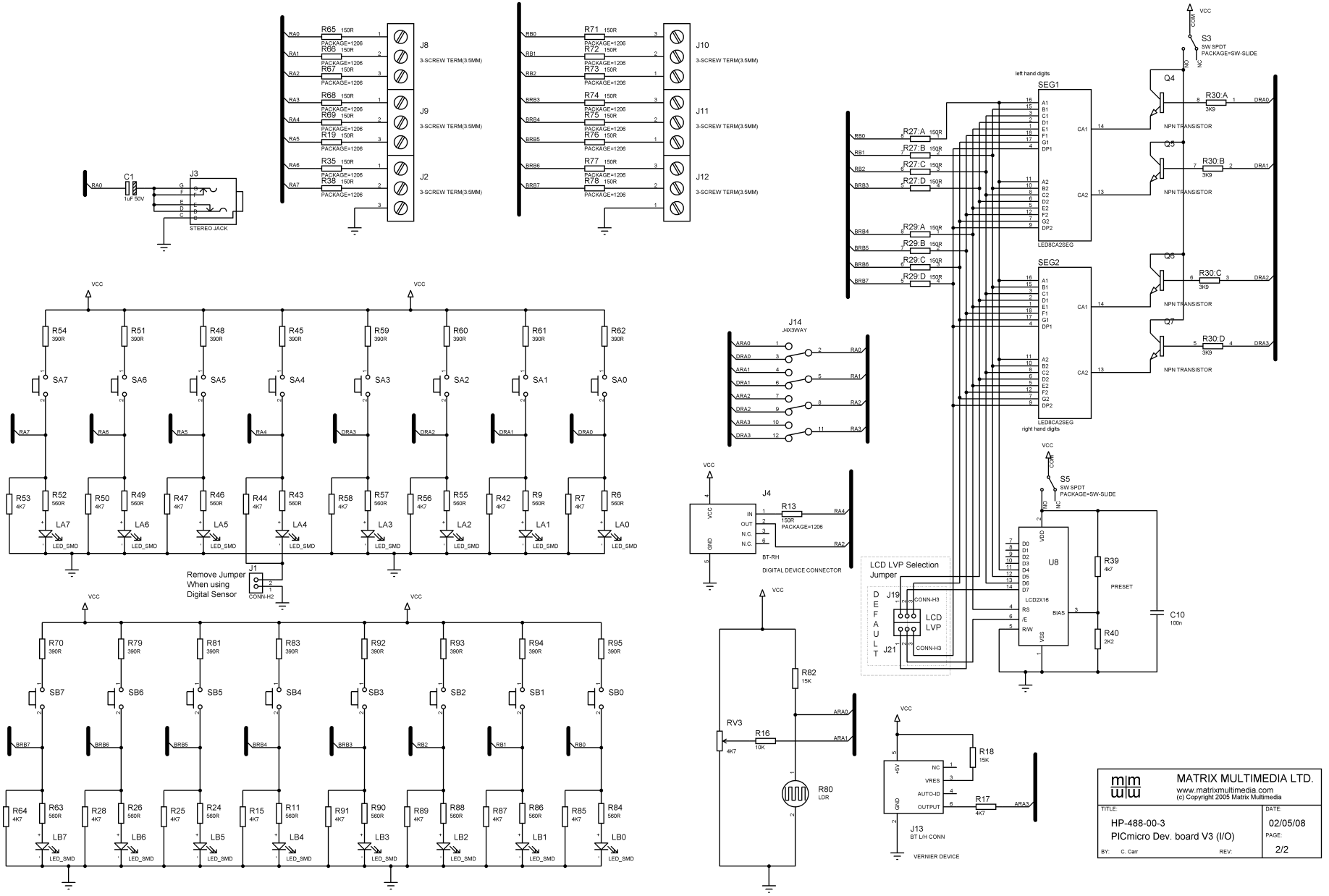
3. Core sensors

We recommend that customers who want to introduce their students to sensors purchase a core set that will introduce students to different techniques and strategies of gathering real world data in both analogue and digital form. In brief these are:

Sensor	Output/action	Coding strategy
Temperature probe	Simple potential divider	A/D conversion, calibration, value look up, display
Motion detector	Gives out a digital pulse correlating to distance	Active sensor, pulse time measurement, conversion and display
Heart rate monitor	Gives out an analogue voltage pulse	Data slicing, and timing
Photogate and pulley wheel	When IR light path is interrupted, digital output changes	Various

Appendix 1 – Circuit diagram

Appendix 1 – Circuit diagram



Appendix 1 – Circuit diagram